

AD-A091 840 DEFT LABS INC EAST SYRACUSE NY
IMAGE ALIGNMENT AND CORRELATION SYSTEM. (U)
JUL 80 S E CRAIG, A L MOYER

F/6 9/2

UNCLASSIFIED 0101-A002

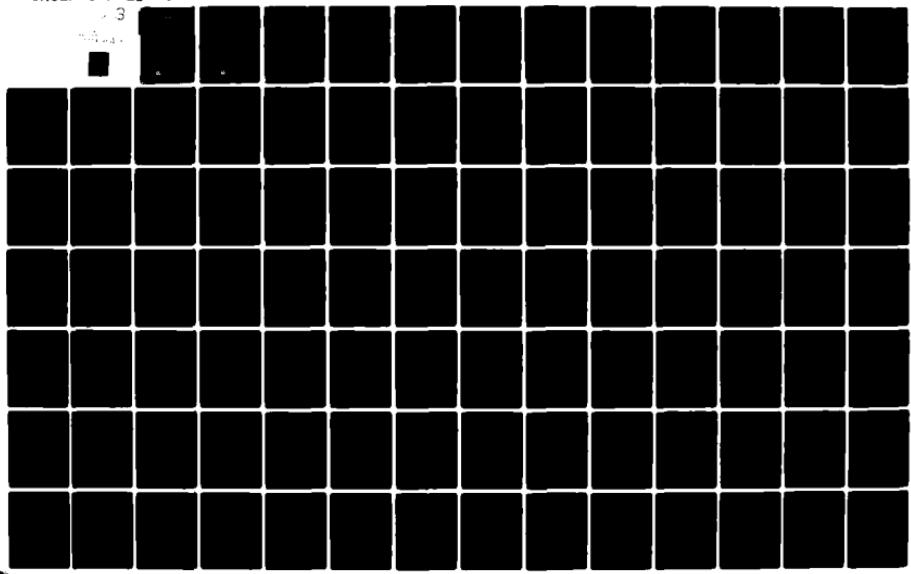
ETL-0237

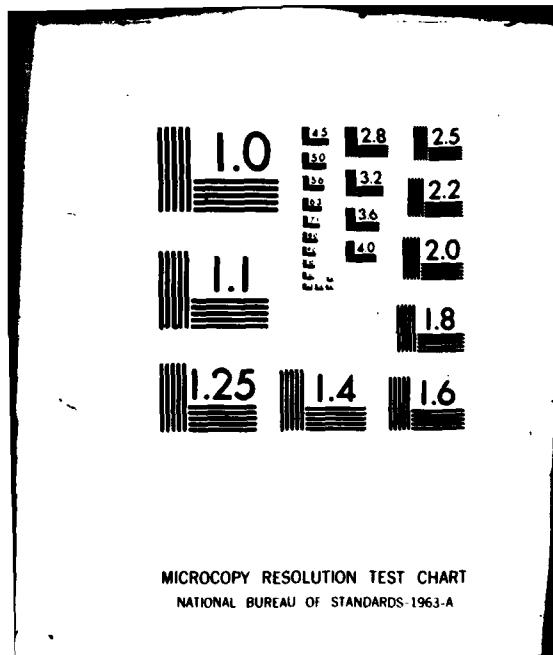
DAAK70-78-C-0217

NL

3

REF ID: A6210





AD A091840

LEVEL

10

18 ETL-0237 19

IMAGE ALIGNMENT AND CORRELATION SYSTEM

10 Samuel E. Craig
Alan L. Moyer

14 D101-A011

DTIC

REPRODUCED

NOV 7 1980

11 July 1980

12 194

9 Final Technical Report. Sep 78

C

1 Jan 80

15 DAAK70-78-1-0217

Prepared for

U.S. Army Engineer Topographic Laboratories
Fort Belvoir, VA 22060

16 4A161102B52

Approved for public release; distribution unlimited

DDC FILE COPY



Deft Laboratories Inc.
7 Adler Drive, East Syracuse, N.Y. 13057

112056

80 11 05 001

ETL-0237

IMAGE ALIGNMENT AND CORRELATION SYSTEM

Samuel E. Craig

Alan L. Moyer

July 1980

Final Technical Report

Prepared for

U.S. Army Engineer Topographic Laboratories
Fort Belvoir, VA 22060

Approved for public release; distribution unlimited



Deft Laboratories Inc.
7 Adler Drive, East Syracuse, N.Y. 13057

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM.
1. REPORT NUMBER ETL-0237	2. GOVT ACCESSION NO. AD-A091 840	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and subtitle) IMAGE ALIGNMENT AND CORRELATION SYSTEM	5. TYPE OF REPORT & PERIOD COVERED Final Technical Report Sept 78 through July 80	
7. AUTHOR(s) Samuel E. Craig Alan L. Moyer	6. PERFORMING ORG. REPORT NUMBER 0101-A002	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Deft Laboratories, Inc. 7 Adler Drive East Syracuse, New York 13057	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 4A161102B52C	
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Engineer Topographic Laboratories Research Institute Fort Belvoir, VA 22060	12. REPORT DATE July 1980	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. NUMBER OF PAGES 194	
15. SECURITY CLASS. (of this report) Unclassified		
16. DECLASSIFICATION/DOWNGRADING SCHEDULE		
18. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
19. SUPPLEMENTARY NOTES		
20. KEY WORDS (Continue on reverse side if necessary and identify by block number) Image Processing, Acousto-Optics, Microprocessors		
21. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report documents the development and design of the Image and Alignment and Correlation System built for the U.S. Army Engineer Topographic Laboratories by Deft Laboratories, Inc. The purpose of the system is to provide a hardware demonstration of the applicability of DEFT (Direct Electronic Fourier Transform) technology to the problems of image		

20. ABSTRACT (cont.)

alignment and image cross-correlation measurement. These problems are related generally to the areas of topographic mapping, feature extraction and change detection, and photo-interpretation.

The system uses a highly developed image-adaptive alignment algorithm which exploits the spatial frequency analysis capability of the DEFT sensor. With high-contrast images having prominent spatial frequencies, residual alignment errors are typically 50 microns in translation and 0.1 degree in angle. The system also has the capability of displaying the spatial frequency content of an image, and of computing normalized cross-correlation coefficients based on spatial frequency data.

The major limitations of the system are its slow operating speed, which is caused by certain parts of the circuitry rather than the sensor, and its dependence on the image.

Accession For	
NTIS GRAAE	
DTIC TAB	
Unannounced	
Justification	
By	
Distribution/	
Availability Month	
Dist	Av. 1970 Dec Special

UNCLASSIFIED

SUMMARY

This report documents the development and design of the Image and Alignment and Correlation System built for the US Army Engineer Topographic Laboratories by Deft Laboratories, Inc. The purpose of the system is to provide a hardware demonstration of the applicability of DEFT (Direct Electronic Fourier Transform) technology to the problems of image alignment and image cross-correlation measurement. These problems are related generally to the areas of topographic mapping, feature extraction and change detection, and photointerpretation.

The development of this system is a continuation of the advancement of DEFT technology and its applications. The sensor technology has received previous sponsorship at Syracuse University by ETL and by the Night Vision Laboratories. The Image Alignment and Correlation System represents a significant achievement in the application of the technology and in its interfacing and programmable control by a microprocessor. In other words, the system represents the first real "use" of DEFT sensor outputs.

The system uses a highly developed image-adaptive alignment algorithm which exploits the spatial frequency analysis capability of the DEFT sensor. With high-contrast images having prominent spatial frequencies, residual alignment errors are typically 50 microns in translation and 0.1 degree in angle. The system also has the capability of displaying the spatial frequency content of an image, and of computing normalized cross-correlation coefficients based on spatial frequency data.

The system consists of two major assemblies, which are the alignment fixture and the electronics cabinet. The alignment fixture uses translation and rotation stages driven by stepper motors to align the test image with respect to the reference image. The images are transparencies mounted on light boxes.

An x-y plotter and a graphics terminal were also furnished with the system as accessories.

The major limitations of the system are its slow operating speed, which is caused by certain parts of the circuitry rather than the sensor, and its dependence on the image.

PREFACE

This report contains all information specific to the Image Alignment and Correlation System, but it does not include background material on the principles of operation and characteristics of the DEFT sensor on which the system is based. That information can be obtained from any one of a number of previous papers and articles, the most significant of which are listed in a bibliography at the end of this document.

TABLE OF CONTENTS

	<u>Page No.</u>
Summary	iii
Preface	v
List of Figures	viii
List of Tables	ix
I. Introduction	1
II. System Hardware	3
A. General Description	3
B. Microprocessor Subsystem	6
C. Stepper Motors and Controls	10
D. Frequency Synthesizers	13
E. RF Distribution Circuit Board	13
F. DEFT Sensor Modules	15
G. Synchronous Detector Circuit Board	17
H. Data Acquisition System (DAS)	19
I. Bias Generator	20
J. Power Supplies	20
K. Light Boxes	21
III. Alignment Algorithm	23
A. Overview	23
B. Spatial Frequency Selection	25
C. Angular Alignment	31
D. Least Squares Estimation of ΔX and ΔY	34
IV. System Software	43
A. General Description	43
B. Supervisor Program	43
C. Plotter and CRT Display Routine	45
D. Alignment Program and Its Subroutine	47
E. Calculation of Image Transform Cross-Correlation Function	63
F. Major Shared Subroutines	65
V. Operating Instructions	70
A. Operating Functions	70
VI. Review of the System's Development	81
VII. Conclusion	85
VIII. Recommendations	86
IX. References	87
X. Bibliography	88
Appendix A - Schematic Diagrams	
Appendix B - Processor Assembly Code Listings	

List of Figures

<u>Figure No.</u>		<u>Page</u>
1	System Block Diagram	5
2	RF Distribution Block Diagram	14
3	Sensor Module Block Diagram	16
4	Synchronous Detector Block Diagram	18
5	Spatial Frequency Search Area	29
6	Overall Software, Flow Diagram	48
7	Spatial Frequency Selection, Flow Diagram	50
8	Coarse Angular Alignment, Flow Diagram	51
9	Fine Angular Alignment, Flow Diagram	52
10	Least Squares Estimation, Flow Diagram	53
11	Subroutine MEASRE, Flow Diagram	54 - 55
12	Correlation Coefficient, Flow Diagram	66

List of Tables

<u>Table No.</u>		<u>Page</u>
1	Memory Address Allocations	11
2	Power Supply Currents	22
3	Typical Residual Alignment Errors	26
4	Coding of Operational Modes	49
5	Pre-Programmed System Functions	70

I. INTRODUCTION

The Image Alignment and Correlation System was built for the U.S. Army Engineer Topographic Laboratories by Deft Laboratories, Inc. under Contract No. DAAK70-78-C-0217. Its purpose is to provide a hardware demonstration of the applicability of DEFT (Direct Electronic Fourier Transform) technology to the problems of image alignment and image cross-correlation measurement. These problems are related generally to the areas of topographic mapping, feature extraction and change detection, and photointerpretation.

The development of this system is a continuation of the advancement of DEFT technology and its applications. The sensor technology has received previous sponsorship at Syracuse University by ETL and by the Night Vision Laboratories. The Image Alignment and Correlation System represents a significant achievement in the application of the technology and in its interfacing and programmable control by a microprocessor. In other words, the system represents the first real "use" of DEFT sensor outputs.

The system performs three main functions: display, image alignment, and image correlation. There are options under each of these main functions. That of display uses as an output device either a graphics terminal which has a storage cathode-ray tube (CRT) or an x-y recorder for "hard copy." The sensor output is scanned in the spatial frequency domain under microprocessor control and displayed graphically in pseudo-three-dimensional form.

The second function aligns a test image in angle and in vertical and horizontal translation, either with respect to a reference image viewed by a second sensor, or with respect to the original position of the test image using only one sensor. Using high-contrast images which have prominent features in the spatial frequency domain, residual alignment errors are

typically 50 microns in translation and 0.1 degree in angle.

The system measures image correlation by computing a normalized cross-correlation coefficient between the outputs of the two sensors over the usable range of the spatial frequency domain. The data on which the correlation coefficient is based is an approximation to the two-dimensional Fourier transform of each image's intensity pattern. Either a real coefficient using the data magnitudes or a complex coefficient using the complex data can be computed.

Because the entire system is controlled by a microprocessor, there is ample opportunity for program modification or expansion. In addition, the system can be used as a stand-alone microcomputer, either with its own terminal or with another.

The purpose of this report is to document the development and design of the system, describing both its hardware and software aspects in detail. The following sections begin with descriptions of the system hardware, the basis of the alignment algorithm, and the software which implements it and the other functions. These descriptions are followed by instructions for operating the system which are an expansion of the Condensed Operator's Manual furnished when the hardware was delivered. Then there is a review of the significant events, technical problems, and their solutions which were experienced during the contract term.

II. SYSTEM HARDWARE

A. General Description

The Image Alignment and Correlation System consists of two major assemblies and two accessories. The two major assemblies are an alignment fixture and an electronics cabinet. The two accessories are a display terminal and an x-y plotter. Some of the system functions do not require the accessories, and they can be detached and used for other purposes.

The components of the alignment fixture are mounted on an L-shaped frame made of 0.5 in. thick aluminum plate. The vertical end of the frame supports a fixed mounting for the reference image and tandem-mounted translation and rotation stages which hold the test image. Each image is a transparency, and is placed on a light box. The translation and rotation stages are driven by stepper motors which are controlled by circuitry in the electronics cabinet. At the other end of the alignment fixture is a bracket which supports a pair of modules containing the DEFT sensors.

Three cables connect the alignment fixture to the electronics cabinet. One carries signals and power to the DEFT modules, and one carries power to the three stepper motors. The remaining cable is an ac line cord for the light boxes.

The electronics cabinet contains circuits which pulse the stepper motor windings, a microprocessor which controls the entire system, some analog signal processing circuitry, and power supplies.

The three stepper motor drivers are located in the center section of the cabinet. Each driver has its own power supply for itself and for the corresponding motor. The drivers are interfaced to an input/output port for the microprocessor. During image alignment the stepper motors are controlled

automatically through this interface. However, they can also be actuated manually by means of push-button controls on the front panel of the cabinet.

The top section of the cabinet front panel is a door which allows access to the microprocessor and the signal processing components. Both are assembled on plug-in circuit cards. Normal operation of the system does not require access to these cards, so the front panel door may be left closed. On the door is a 16-position keypad and a 15-position LED display. The keypad is used to select and execute the various functions which the system has been programmed to perform. The LED display shows either the positions of the translation and rotation stages or the value of the correlation coefficient, depending on the program selected. It also displays the number of the selected program and gives error warnings under certain conditions.

The bottom section of the electronics cabinet contains power supplies for the microprocessor, the analog signal processing circuits, and the DEFT sensor modules. Two power switches are located on the front panel of this section. The one on the left side is a master switch for the entire system except for the light boxes. The other switch controls the light boxes independently.

The rear panel of the cabinet is a door on which are mounted the connectors for the interfacing cable to the DEFT sensor modules, the stepper motors, and the accessory x-y plotter and display terminal. All of these connectors are different, so there is no danger of connecting the interfaces improperly. Also located on the rear panel door are the ac line fuse and two small toggle switches. These switches were used for tape cassette read/write operations when the system software was being developed. They have no effect on the presently programmed system functions.

Figure 1 shows a block diagram of the entire system. The

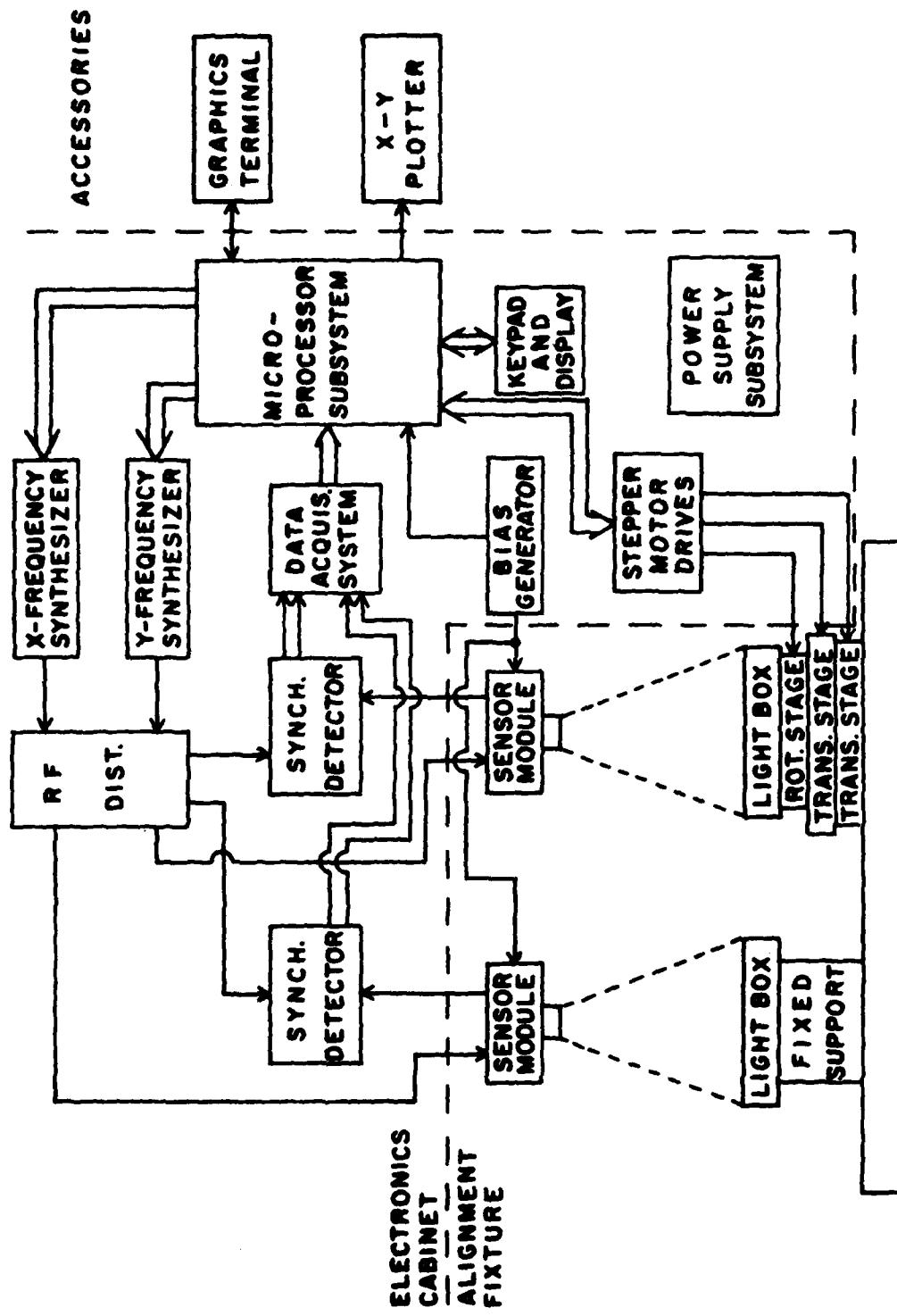


Figure 1 - Functional Block Diagram of Image Alignment and Correlation System

signal flow is generally from the synthesizers, which are controlled by the microprocessor, to the rf distribution circuit board, and from there to the sensor modules and synchronous detector circuit boards. Signals from the modules go to the synchronous detectors and then to the data acquisition system where they are sampled and digitized synchronously under software control. The bias generator is coupled to the microprocessor to achieve this synchronization. The microprocessor also controls the stepper motors through their drivers and the accessory x-y recorder through the analog output. It also exchanges data with the keypad and LED display and with the accessory graphics terminal. The power supply subsystem furnishes power to all parts of the system except the accessories, the stepper motors and their drivers.

B. Microprocessor Subsystem

1. General Description - The microprocessor subsystem consists of an integrated collection of modular plug-in circuit cards purchased from Wintek Corp. of Lafayette, IN. The modular approach allows the processor to be configured to meet the needs of the system at minimum cost. In addition, Wintek offers a very useful monitor in firmware called FANTOM-II, as well as an editor/assembler, which enabled the same processor to be used as a software development tool. The processor is an 8-bit machine based on the 6800, and includes a Control (CPU) Module, a ROM module with 16K capacity, a 16K dynamic RAM module, a RAM refresh module, an analog output module, and a two-port parallel I/O module. All of these cards are held in a cage behind the front door of the electronics cabinet. In addition, there is a Console I/O Module with a keypad and LED display which is mounted on the door.

All of the connections between the microprocessor and the remainder of the system, with the exception of the Data Acquisition Subsystem (DAS), are made through 6820/6821 Peripheral Interface Adapters (PIA's). The DAS is connected

directly to the address and data busses through buffers located on an auxiliary card. All peripherals are memory-mapped since the 6800 has no IN or OUT instructions; that is, the peripherals are treated in the same way as memory locations by the processor. The address decoding scheme and other details can be found in the Wintek information which is included separately with the Commercial Data.

2. Control Module - This card includes the 6800 CPU, a 1K ROM which contains the FANTOM-II monitor, $\frac{1}{2}$ K of RAM, two PIA's, and a 6850 Asynchronous Communications Interface Adapter (ACIA). The ACIA is configured to provide transmission and reception of ASCII data through an RS-232 interface with the accessory graphics terminal. The two PIA's provide two groups each of 16 lines of parallel output data which controls the two frequency synthesizers. The interconnection is made through a 50-conductor ribbon cable.

3. Cassette Interface - This module does not plug into the card cage, but rather is mounted on the inside of the rear cabinet door. It is connected to the Control Module by a 14-conductor ribbon cable, and converts logic voltage levels to RS-232 levels and vice versa. It also can act as a modem for writing data to a cassette tape recorder and reading it back into memory later. The audio input and output lines are brought out to the connector which interfaces the system with the x-y plotter, since the plotter and the cassette recorder would not be used simultaneously. The cassette functions were used extensively while the software was being developed. They remain functional, but they are unnecessary in normal system operation. Cassette read and write procedures can be found in the Wintek documentation with the Commercial Data.

The small 7-position DIP switch on the Cassette Interface circuit board sets the baud rate for the RS-232 interface. This setting must agree with the rate set at the accessory graphics terminal. The recommended rate is the maximum accepted by the

terminal in order to minimize execution times. In the case of the Tektronix 4006-1 terminal this rate is 4800 baud. To set this rate, position 6 on the DIP switch should be ON. All other positions should be OFF.

The small toggle switches on the rear panel affect only the cassette read and write operations.

4. RAM and Refresh Modules - the RAM module holds 16K bytes of dynamic read-write memory. Refreshing of this memory, along with some address decoding functions, is done by the RAM Refresh Module. These two cards are interconnected at their front ends as well as at the backplane.

A portion of this 16K block is unused by the existing software and is available for the temporary storage of other programs or data. In addition, the system can accommodate another 16K module without any hardware or software changes.

5. ROM Module - This card has the capacity for sixteen 2708-type UV-erasable programmable read-only memory (EPROM) chips. As delivered, only nine sockets are in place. Eight of them contain the 8K bytes of permanent software for the image alignment, correlation, and plotting functions. The ninth EPROM was purchased from Wintek and is programmed with a set of math routines which are used by the other software.

This module also can be expanded to its capacity of 16K bytes.

6. Analog Output Module - This card has two 8-bit digital-to-analog (D/A) converters which furnish the x and y input voltages for the x-y plotter. These voltages are taken to the rear panel connector through a ribbon cable which plugs into the front of the card. The D/A converters are adjusted for an output of 0.00V for a binary input of 00000000, and an output of +10.00V for an input of 11111111 (FF_H).

7. Parallel I/O Module - This unit has a PIA and buffers, and provides two 8-bit parallel ports, one input and one output,

for interfacing with the stepper motor drivers. These interconnections are made by means of two ribbon cables.

Although this module has the capacity for four PIA's, only one more can be added without creating a memory address conflict, because of the address decoding scheme used by Wintek.

8. Console I/O Module - This module is mounted on the front panel of the cabinet so that its 16-position keypad and 15-position LED display can be seen and accessed easily. Its edge connector is wired to the backplane using only those lines necessary for its operation. The system software recognizes commands entered at the keypad and uses the display to show the number of the program being (or about to be) executed. It also shows the relative position of the test image and the value of the correlation coefficient when appropriate.

This circuit generates interrupts, which are maskable, at a rate of 1200 per second. The interrupt service routine updates the display and checks the keypad for input. Further details are given in the software description and with the Commercial Data.

9. Reset Generator - An auxiliary circuit card, located adjacent to the left end of the microprocessor backplane, contains address and data buffers for the DAS. It also has circuitry which produces a hardware reset when the system power is turned on and when the BREAK key on the graphics terminal is pressed. A hardware reset affects all of the interface adapters and the CPU, and is necessary before the system software can configure the interface adapters for their various functions. It also causes the processor to get an address from a particular memory location and to begin execution at that address. The address is A91CH, the entry point for the initialization procedure. In the case of a system malfunction which stops normal program execution, it will be necessary to reset the

system to regain control of it. Rather than remove power and reapply it, a small white button on the auxiliary card can be used. This button simply grounds the reset line.

When the front cabinet door is opened, two such buttons will be seen on adjacent cards at the left side. The reset button is the leftmost of the two. The other one grounds the non-maskable interrupt (NMI) line, and has the effect of stopping any program execution in progress and giving control to the FANTOM-II monitor. The monitor can be used only through the keyboard of a terminal connected to the RS-232 interface. This capability is useful for troubleshooting or for experimenting with new software, but it will not be needed in normal system operation. If the NMI button should be pressed accidentally, simply press the Reset button to escape from the monitor. A program listing for the monitor and instructions for using it are included with the Commercial Data.

10. Memory Allocations - The addresses for the system's memory and peripheral adapters are given in the following table. All addresses are in hexadecimal notation. Address space which is designated as "Not Available" is such because of the Wintek address decoding scheme which does not use all 16 address bits in all cases. Addresses so designated must be avoided since their use can result in the simultaneous activation of more than one peripheral, with ambiguous results. Addresses not listed in Table 1 in the range $E000_H$ - $EFFF_H$ fall in this category, with two exceptions. Either $EE04$ - $EE07$ or $EF04$ - $EF07$ (but not both) may be used without any hardware modification. This space will accommodate either one additional PIA (on the Parallel I/O Module or on a separate module) or up to two additional ACIAs without conflict. See Wintek application note AN-0010 in the Commercial Data for more detail.

C. Stepper Motors and Controls

The translation and rotation stages and their controllers

TABLE 1

Memory Address Allocations

<u>Hexadecimal Address</u>	<u>Function</u>
0000 - 3FFF	RAM
4000 - 7FFF	Available for future expansion
8000 - BFFF	ROM
C000 - DEFF	Available for future expansion*
DFX0 - DFX7	Available for future expansion
DFX8 - DFXF	Data Acquisition Subsystem
E000 - EBFF	Not Available
EC00 - EDFF	RAM
EE08 - EE09	ACIA
EE10 - EE13	PIA for X Synthesizer
EE20 - EE23	PIA for Y Synthesizer
EE40 - EE43	PIA for Front-Panel Console
EE80 - EE83	PIA for Analog Output
EF00 - EF03	PIA for Stepper Motors
F000 - FBFF	Not Available
FC00 - FFFF	ROM (FANTOM-II Monitor)

*With additional address decoding hardware.

Note - See text regarding addresses starting with E.

were purchased from Aerotech, Inc. of Pittsburgh, PA. Each stage is actuated by a small stepper motor and each motor is controlled by a driver. Aerotech calls the driver a "translator," probably because it translates logic-level signals or switch closures into voltages which are applied to the motor windings in the proper sequence. Each driver is connected to one of the front-panel push-button switch assemblies by a ribbon cable. The upper switch assembly operates either the horizontal (x) or the vertical (y) translation stage, while the lower set of switches operates only the rotation stage.

Instructions and schematics pertaining to the driver (or "translator") and the switch assembly are included in the Commercial Data which is separate from this volume.

The stepper motor subsystem is interfaced to the microprocessor through two 8-bit ports on its Parallel I/O Board. One port is configured as an output and furnishes clock and direction signals to Pins 3 and 4, respectively, of J1 on each of the driver ("translator") boards. Since there are three stepper motors, only 6 of the 8 bits are used. Separate outputs which confirm the clock and direction signals are taken from J1 on the front-panel switch assemblies to the other I/O port, which is configured as an input. Again, only 6 of the 8 bits are used.

When the local/remote switch (labeled MAN/AUTO) is in the local (MAN) position, the outputs from the microprocessor are disconnected from the motor drivers. The drivers then generate their own clock pulses under the control of the STEP and SLEW switches, and the direction switch determines the motor direction. With the local/remote switch in the remote (AUTO) position, the STEP, SLEW, and direction switches have no effect and the stepper motors operate under the control of the microprocessor. The driver outputs which confirm the clock and direction signals remain connected to the processor in both modes so that it can keep track of the relative position of

each stage when the system is in its calibrated state.

Each driver board has its own power supply.

D. Frequency Synthesizers

Two digital frequency synthesizers are used in the system. They ultimately provide the excitation for the orthogonal surface-acoustic wave (SAW) transducers on the DEFT sensors, and they also furnish the reference for synchronous detection of the sensor outputs.

The synthesizers are modular plug-in circuit boards made by Syntest Corp. of Marlboro, MA. They are capable of covering the range of 20.000 MHz to 159.999 MHz in 1 kHz steps, and are controlled by the parallel input of 5½ BCD digits at logic levels. In this system the most and least significant digits are always zero because they are hardwired to ground. The remaining four BCD digits for each synthesizer come from the two PIA's on the Control Module. The resulting effective frequency range is 20.00 MHz to 99.99 MHz in 10 kHz steps, and each synthesizer is controlled independently.

The two synthesizers use a common crystal-controlled 1 MHz reference which is located on one of them and cross-connected to the other. The synthesizer without the reference oscillator cannot function without the other one.

The synthesizers are located immediately to the right of the microprocessor back plane. The x-frequency is produced by the one on the left. Their outputs are carried to the rf distribution circuit board by miniature 50-ohm coaxial cables.

Specifications and other information on the synthesizers can be found with the Commercial Data.

E. RF Distribution Circuit Board

Figure 2 shows a block diagram for this unit. It is a plug-in card located immediately to the right of the synthesizers.

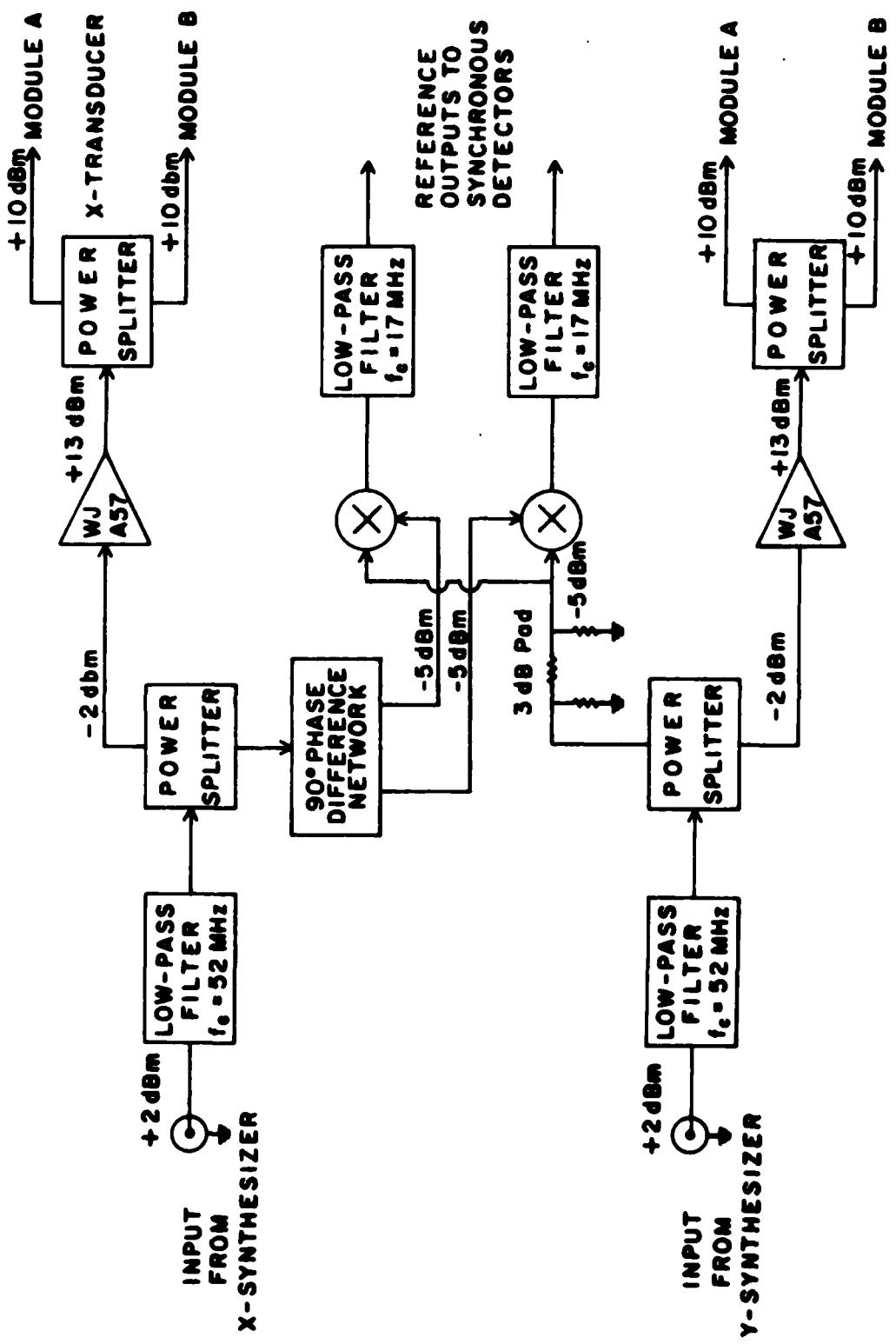


Figure 2 - Block Diagram of RF Distribution Circuit Board

Its function is to filter and amplify the synthesizer outputs and forward them to the DEFT sensor modules. In addition, the x-axis frequency is split into two orthogonal parts in a 90-degree phase difference network and each part is mixed with the y-axis frequency in a balanced modulator. The difference frequency is isolated by a low-pass filter at the output of each balanced modulator. The resulting difference-frequency signals, which are in phase quadrature, are sent on to the synchronous detector cards.

F. DEFT Sensor Modules

The two sensor modules are located on the alignment fixture. They have identical circuitry, but there are minor differences in the characteristics of the sensors themselves. The system software takes these differences into account, and for that reason the sensor modules should not be interchanged.

Figure 3 shows a block diagram for the sensor modules. The x-axis and y-axis frequencies from the rf distribution circuit board are applied to the SAW transducers on the sensor through drivers and matching networks. The purpose of the matching networks is to increase the effective acoustic bandwidth of the transducers.

Before this system was built, the practice was to apply a dc bias to the contact pattern on the sensor, and the resulting output appeared at a frequency equal to the difference between the two SAW frequencies. However, in this design, a sinusoidal ac bias is used and the output takes the form of a double-sideband suppressed carrier signal. The sidebands are displaced from the difference frequency by the bias frequency, which is 1440 Hz in this system. This scheme provides a spectral separation of the desired signal from any component at the difference frequency which might result from stray mixing of the two transducer voltages, since they are very large compared to typical signals.

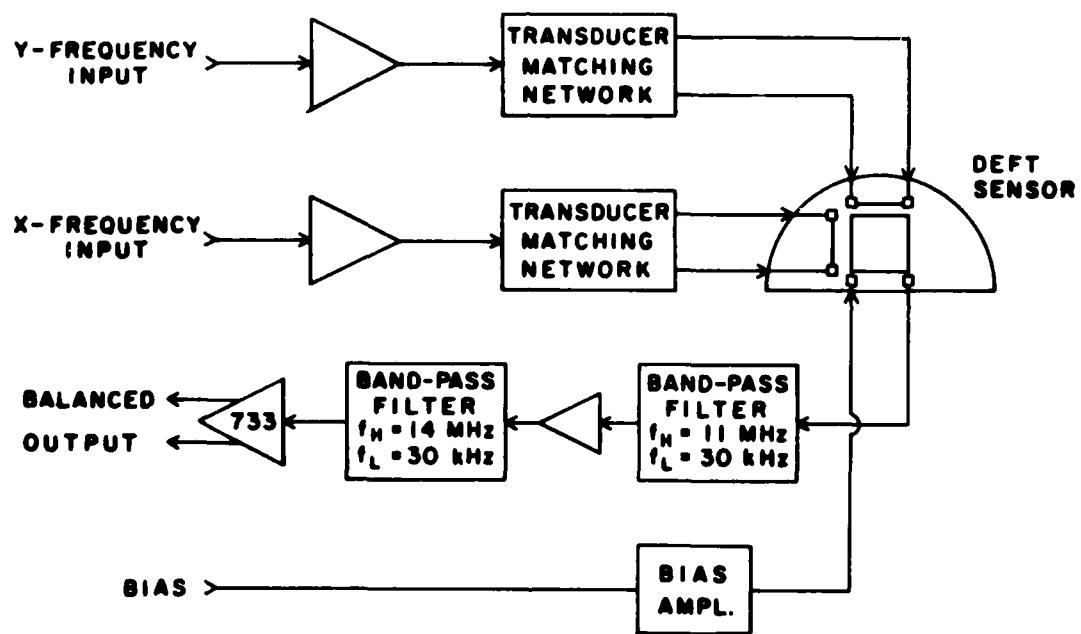


Figure 3 - Block Diagram of DEFT Sensor Module

A low-pass filter between the sensor and the signal pre-amplifier attenuates frequencies above the highest difference frequency, which in this case is about 10 MHz. The impedance of the sensor's contact pattern is mostly capacitive and forms an integral part of the filter circuit. Since the sensor is usually characterized as a current source as far as the signal is concerned, the preamplifier can be thought of as a current-to-voltage converter with a transimpedance of about 2.5K. Its output is again low-pass filtered and further amplified. The output of the second amplifier stage is balanced and each side is matched to the 50Ω coaxial cables which carry the signal to the electronics cabinet.

The disc behind the lens on the module can be rotated to reveal a peep-hole on each side so that the image on the sensor can be seen. The lens is a standard one-inch format CCTV type with a "C" mounting thread (1"-32).

G. Synchronous Detector Circuit Board

The signal from each sensor module is fed to a synchronous detector card in the electronics cabinet. There are two such cards, one for each sensor module, and they are located immediately to the right of the rf distribution circuit board. They are readily identified by the 26-conductor ribbon cable which connects to the front of both of them.

Figure 4 shows a block diagram for the synchronous detector board. It consists of a pair of balanced modulators to which the signal from the sensor is fed in parallel. The reference inputs to the modulators are the difference-frequency voltages in phase quadrature which come from the rf distribution board. The balanced modulators translate the double sideband suppressed carrier signal to the bias frequency, which is constant. Low-pass filters which follow the modulators essentially remove all other frequencies from the outputs. These two bias-frequency signals are in phase, but their voltages represent two orthogonal

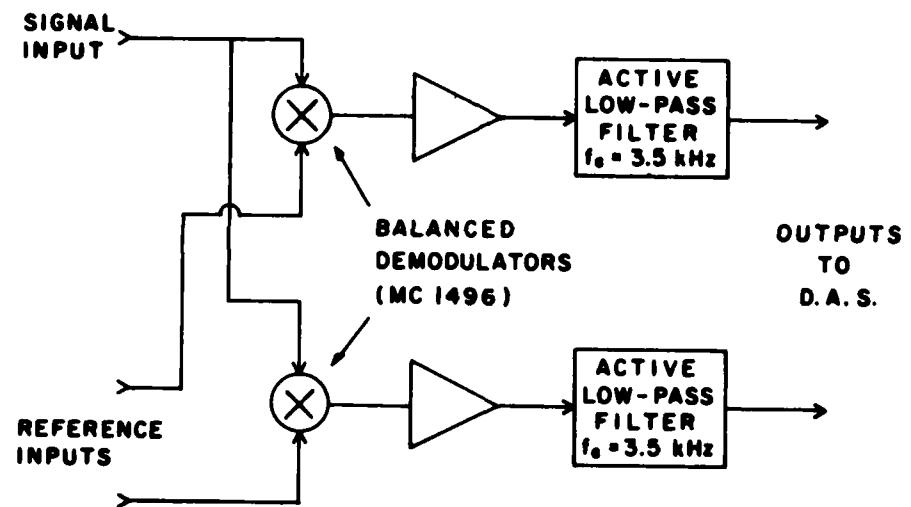


Figure 4 - Block Diagram of Synchronous Detector Circuit Board

components of the signal from the sensor. These two voltages connect to the DAS via the ribbon cable and are digitized synchronously under software control.

The synchronous detector cards have other circuitry which is not used in the present system. It was included in the original design of the card prior to the decision to implement a synchronous detector in software for the bias-frequency signal.

H. Data Acquisition System (DAS)

The DAS is in the form of a plug-in circuit card purchased from Analog Devices of Norwood, MA. Wintek offers an input version of their analog interface module but it was considered to be too slow for this application. No source was found for an analog input module compatible with the Wintek bus. The Analog Devices model RTI-1220 was selected, but buffers for the address lines and for four bits of the data bus had to be included on an auxiliary card to prevent excessive loading of the Wintek bus.

The DAS consists of an input multiplexer, a sample-and-hold circuit, a 12-bit A/D converter, and control logic. The input is configured for 16 channels of analog data with a common reference for "pseudo-differential" operation. Only four channels are used at present, and since their sources are the two side-by-side synchronous detector cards, negligible error is introduced by using a common reference. The selection of the input channel and the sampling of the analog voltage are both under software control. A conversion is initiated automatically when the sample-and-hold goes to the "hold" mode. This module is set for 2's complement binary output with an analog input range of $\pm 5V$.

A data sheet for this unit is included with the Commercial Data. More detailed information is contained in the User's Guide for the RTI-1220 and RTI-1221, which was supplied separately because of its copyright restriction.

I. Bias Generator

The bias generator furnishes the ac bias voltage for the DEFT sensors. It is the circuit card at the extreme right end of the cage. It consists of a phase-locked loop frequency synthesizer using the 60 Hz line as a reference. A miniature rotary switch at the end of the circuit card sets the output frequency, which is equal to the switch setting multiplied by 240 Hz. The switch should remain set to 6 for a bias frequency of 1440 Hz, since the timing in the software synchronous detector is matched to that frequency.

Logic-level square waves derived from the 60 Hz line and from the bias output are made available to the microprocessor through extra inputs to the PIA's on the Control Module. The synchronous detector subroutine uses these signals for timing references. The 60 Hz waveform is obtained from a small filament transformer.

The bias voltage applied to the sensors has a peak value of about 5V.

An LED below the frequency-setting switch indicates loss of phase lock. Normally it will flash when the system is turned on or if the frequency setting is changed. Otherwise it should remain off at all times.

J. Power Supplies

The power supply subsystem at the bottom of the electronics cabinet was procured from Acopian Corp. of Easton, PA. and is identified by their number 3276. The chassis wiring was modified slightly to make some of the rear terminals available for the switch which was added for the light boxes.

This subsystem consists of four separate modular supplies. One furnishes +5V and $\pm 12V$ to the microprocessor system. These three outputs are overvoltage protected. The +5V also goes to the DAS. Another supply provides $\pm 15V$ for the DAS, the bias generator, and all of the analog circuitry, which includes the

DEFT modules and the rf distribution and synchronous detector circuit boards. The remaining two units have outputs of +9V and +24V, respectively, for the frequency synthesizers.

The following table lists each available voltage, the rated maximum current available at that voltage at an ambient temperature of 60C, and the measured current drain with the system operating. It is evident that all of the ratings are conservative.

K. Light Boxes

Illumination for the test and reference images is provided by light boxes on which the images are held. The illuminated area is about 11 inches square, and images should be transparencies of approximately that size. They can be kept flat by squares of anti-reflective glass which were furnished with the system. The glass covering the image is retained by four spring fasteners on the light boxes.

The light source in each box is an array of six F6T5/CW fluorescent lamps, each having its own starter and ballast. Access to these parts is gained by removing the 24 screws which hold the translucent plastic cover to the box. Before removing the cover, it should be marked temporarily (with a piece of tape) so that it can be replaced with the same orientation.

A single power cord for both light boxes comes from the box which is fixed in position. This cord has a standard 3-prong grounding plug which can be mated to the receptacle on the rear door of the electronics cabinet or to some other 115V 60Hz source if more convenient. The receptacle on the cabinet is switched from the front panel.

The power connection between the two light boxes also has a standard 3-prong disconnect, but the female part has been modified so that the plug can be pulled out easily. This measure was taken in case the rotating stage should accidentally

wrap the short cord around itself. In that event the plug and socket will disengage before tension on the cords causes any damage to the internal connections.

TABLE 2

Measured vs. Rated Power Supply Currents

<u>Voltage</u>	<u>Current Rating (@ 60C), A</u>	<u>Measured Current, A</u>
+5	6.0	2.26
+12	1.2	0.36
-12	1.2	0.30
+15	0.85	0.29
-15	0.85	0.61
+9	2.6	1.3
+24	0.75	.006

III. ALIGNMENT ALGORITHM

A. Overview

The function of the alignment algorithm is to align two identical or nearly identical images (e.g., stereo pairs) which are misaligned in both angle and in translation.

Alignment is achieved by the following steps:

1) Selection of prominent spatial frequencies in the reference image which are then used exclusively in the subsequent processing.

2) A coarse angular alignment based on maximizing the correlation of the magnitude of the two-dimensional Fourier transforms of the reference and misaligned images sampled at the selected spatial frequencies.

3) A fine angular alignment based on maximizing the same correlation function as in 2) above. The final fine search increment is 0.1 degree.

4) A translational alignment based on a least squares estimate of Δx and Δy , the x and y-axis misalignment. Fourier transform phase data is used in this step.

5) A fine angular correction.

Steps 4) and 5) are repeated iteratively until the computed translational correction becomes less than a small threshold.

The alignment algorithm is based on the well-known Fourier transform space-shifting theorem¹. The theorem can be stated simply as follows: Let $I(x, y)$ be the intensity function of the reference image and $I_{\Delta x, \Delta y, \Delta\theta}(x, y)$ be the intensity function of an identical image translated by Δx , Δy and rotated by $\Delta\theta$. Let $F(\omega_x, \omega_y)$ and $F_{\Delta x, \Delta y, \Delta\theta}(\omega_x, \omega_y)$ be the Fourier transforms of these images respectively. Then

$$|F(\omega_x, \omega_y)| = |F_{\Delta x, \Delta y, \Delta\theta}(\omega_x, \omega_y)| \quad (1)$$

and

$$\arg\{F(\omega_x, \omega_y)\} - \arg\{F_{\Delta x, \Delta y, \Delta\theta}(\omega_x, \omega_y)\} = \omega_x \Delta x + \omega_y \Delta y \quad (2)$$

¹J. W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill 1968, chap. 2.

The first equation states that if the two images are in angular alignment (but not necessarily translational alignment) then the magnitude of their Fourier transforms would be identical. Hence, a cross-correlation of the magnitudes of the two transforms achieves its maximum when the two images are aligned in angle. This property is the basis of steps 2), 3) and 5) in the algorithm.

Once the images are in angular alignment the second equation above states that the difference between the phase of the two transforms is a bilinear function of the misalignment Δx and Δy . These offsets could, in principle, be determined using only two spatial frequencies. However, a superior approach is to increase the number of samples and estimate Δx and Δy using least squares estimation. Since a number of samples are used in computing the estimate, the effect of noise at each sample is reduced through averaging over the set of spatial frequencies. This approach is used in step 4) of the algorithm.

There is, however, an additional complication not evident in equations (1) and (2). The DEFT sensors each have a 1.27 cm \times 1.27 cm square aperture onto which the images must be focused. To maximize resolution, it is desirable that $I(x, y)$ completely fills the aperture of one sensor. Since magnification is the same for both images, $|F|$ cannot equal $|F_{\Delta x, \Delta y, \Delta \theta}|$ unless $\Delta x = \Delta y = \Delta \theta = 0$ since, otherwise, part of $I_{\Delta x, \Delta y, \Delta \theta}$ will fall outside the aperture of the second sensor. Hence, equations (1) and (2) are only approximately true. The alignment algorithm has been designed to be insensitive to this approximation. This is accomplished by iterating steps 4) and 5). That is, corrections are computed assuming that equations (1) and (2) are exactly satisfied. These corrections are applied to the misaligned image to bring it into alignment with the reference. Since the equations are only partially satisfied there will be a residual error. New corrections are recomputed and applied

iteratively until the magnitude of the correction falls below a threshold. At that point the two images are assumed to be aligned. If, indeed, the algorithm converges to alignment then, in the limit, equations (1) and (2) will hold exactly.

Analytic conditions for convergence are image dependent, difficult to derive and probably not useful in practice. However, in experiments using test patterns containing prominent spatial frequency components, (e.g., grid patterns), the algorithm was successful in aligning images to high accuracy. Some experimental results are presented in Table 3. The reference image in this case was a black and white checkerboard pattern with a horizontal frequency of 10 line pairs across the sensor aperture and with a vertical frequency of 5 line pairs. The reference image was placed in three orientations: 1) 10 line pairs horizontal (0 degrees), 2) 10 line pairs with a 6.5 degree tilt and 3) 10 line pairs vertical (90 degrees). The misaligned image was identical. The initial and final offsets are shown in the table. For this pattern all final errors were less than 0.1 mm in Δx , Δy with one exception and 0.2 degrees in $\Delta\theta$. (These errors are referred to the light table. Because of the 20:1 demagnification from light table to sensor, the errors Δx , Δy referred to the sensor were all less than 5 microns.)

A detailed description of the alignment algorithm is contained in Sections III B. through III E. Section III F. contains a discussion of the computational requirements of the algorithm. It is shown there that the algorithm has computational advantages over algorithms which use image intensity data as input rather than the Fourier transform.

B. Spatial Frequency Selection

For computation time and signal-to-noise considerations a small number of spatial frequencies must be automatically selected by the program from the large number of addressable spatial frequencies within the bandwidth of the DEFT sensors.

TABLE 3

Initial and Final Offsets: Typical Test Pattern

Reference Orientation	Initial Offsets			Final Offsets		
	Δx (mm)	Δy (mm)	$\Delta \theta$ (mm)	Δx (mm)	Δy (mm)	$\Delta \theta$ (mm)
0	+5.00	+5.00	-2.0	.00	+.04	0.0
0	-5.00	+5.00	-2.0	-.02	+.05	-0.1
0	-5.00	-5.00	-2.0	-.03	-.07	-0.1
0	+5.00	-5.00	-2.0	-.01	-.13	0.0
6.5	+5.00	+5.00	-2.0	+.01	+.03	-0.1
6.5	-5.00	+5.00	-2.0	-.05	+.03	-0.2
6.5	-5.00	-5.00	-2.0	-.06	-.05	-.01
6.5	+5.00	-5.00	-2.0	-.04	-.04	-0.2
90	+5.00	+5.00	-2.0	-.02	+.05	-0.1
90	-5.00	+5.00	-2.0	-.08	+.07	-0.2
90	-5.00	-5.00	-2.0	-.06	+.03	-0.1
90	+5.00	-5.00	-2.0	+.07	+.03	0.0
Mean				-.0183	+.0033	-0.10
Standard Deviation				.0386	.0583	.0577

The method of selection of these spatial frequencies will be discussed in this section.

The alignment algorithm requires the computation of a cross-correlation at each angle increment during angular alignment. In addition, a least squares estimation is required at each iteration of the translational alignment. The computation time required is linearly proportional to the number of spatial frequency samples used. Hence, it is desirable to use as few spatial frequencies as possible. However, since each sample will contain noise due to the sensor, electronics and A/D converter it is necessary to use a set of spatial frequencies so that noise will be averaged out. Through experimentation it has been determined that about 16 spatial frequencies are adequate for the proper functioning of the algorithm.

These spatial frequencies are selected using transform data from the reference image only. A number of criteria are necessary in the selection of these points.

1). The reference transform evaluated at the spatial frequency should have a large magnitude. This increases the signal-to-noise ratio. It is especially important for least squares estimation since noisy phase data is weighted the same as phase data from significant transform components.

2). The set of spatial frequencies must not all lie along a straight line through the origin of transform space. During translational alignment a plane defined by $\omega_x \Delta x + \omega_y \Delta y$ is fitted to data consisting of transform phase differences. Since three points (which are not all on the same line) are required to define a plane, the spatial frequencies selected must not all lie along the same line. In addition, the least squares equation will be ill-conditioned if all data points are clustered along a line through the origin. If the equations are ill-conditioned then a small amount of noise at the A/D output will be magnified to a large error in the computed corrections Δx , Δy . A situation where ill-conditioning would occur would be if all the spatial frequencies selected were

clustered around a single peak in the transform.

3). Spatial frequencies near the origin of spatial frequency space should not be used. The transform magnitude and phase is rather insensitive to translation and rotation for very low spatial frequencies.

4). High spatial frequencies should not be used for coarse translational alignment if Δx or Δy are so large that $|\omega_x \Delta x| + |\omega_y \Delta y| \geq \pi$. This is because true phase cannot be measured. Rather, the principal value of phase is measured. Phase differences greater than π cannot be detected. However, as translational errors are reduced Δx and Δy will be smaller, allowing the use of higher spatial frequencies for fine alignment.

A search scheme which was developed to satisfy these criteria will now be described. The area searched in spatial frequency space is shown in Figure 5. As noted above, a region of low frequencies is excluded. During the first stage of the search, this region is sampled on a grid with a spacing of 210 kHz in both dimensions. Spacing is fine enough to satisfy the Nyquist criteria but coarse enough to require sampling at only 959 points. At each point on this grid the reference transform is sampled and the magnitude of the transform computed. From these magnitudes a table is constructed. The n -th entry in this table is the number of samples for which the magnitude of the sample is greater than $n \times \text{THR2}$ where THR2 is a small constant. (In the software, this table is constructed during calls to subroutine UPDATE.) After the transform is searched, subroutine THRSET is used to determine the number n such that the n -th table entry is less than or equal to NPT and the $n-1$ st entry is greater than NPT . NPT is the desired number of spatial frequencies which is 16. Then the threshold THR1 is set to $n \times \text{THR2}$. The significance of THR1 is the following. If during the first stage of the search, a spatial frequency was accepted if and only if the magnitude of the transform at that frequency was equal to or greater

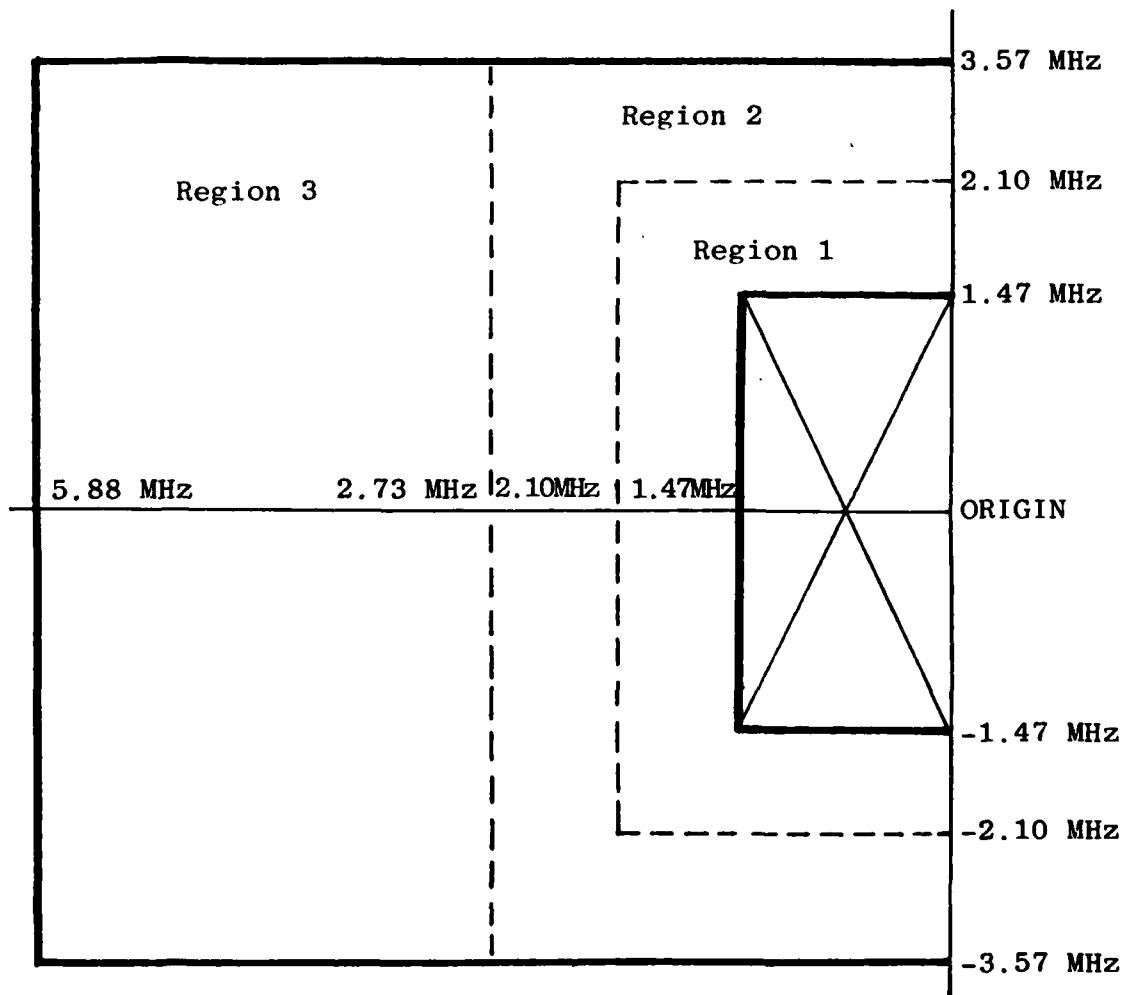


Figure 5 - Search Area in Spatial Frequency Domain

than THR1, then the number of samples accepted would be less than or equal to NPT.

Once THR1 is computed the second stage of the search is initiated. The reference transform is again searched over the coarse 210 kHz grid. However, each time the magnitude of a sample equals or exceeds THR1, a fine search is initiated in the 180 kHz x 180 kHz square centered at the sample. The fine search grid spacing is 30 kHz. (In the software, this search is carried out in subroutine GSRCH.) The spatial frequency with largest transform magnitude in this square is determined. This magnitude is compared with THR1. If it is at least as large as THR1 the spatial frequency is accepted and becomes one of the set to be used in the subsequent operations of correlation and least squares estimation. The use of the fine search results in spatial frequencies with larger magnitudes. Since a fine search is initiated only when THR1 is equalled or exceeded during the coarse search, the time spent in fine search is minimized. This procedure is continued over the entire coarse grid bounded as in Figure 5. Because of the way THR1 was chosen, the number of spatial frequencies chosen will be close to the desired NPT.

Since only one spatial frequency can be chosen in each 180 kHz x 180 kHz square, the set of spatial frequencies chosen tend to represent the prominent frequency components of the transform without clustering exclusively around a single large peak (if the transform contains one.)

To prevent a condition of phase ambiguity as outlined in criterion 4) above, the search area shown in Figure 5 is subdivided into three regions. During least squares estimation of Δx , Δy only spatial frequencies from region 1 are used until the iteration when the computed correction becomes less than a threshold Δ (1). On the next iteration spatial frequencies in both region 1 and region 2 are used. This is the case until the iteration when the computed correction becomes less than the threshold Δ (2). On the next and all succeeding iterations spatial frequencies from all three regions are used.

Hence, as Δx and Δy decrease higher spatial frequencies can be used to provide higher resolution since $\omega_x \Delta x + \omega_y \Delta y$ decreases with Δx , Δy .

In the alignment program the thresholds are stored in a table $\text{DELTA}(L)$. At the beginning of least squares estimation L is set to 1. After a correction is computed, if $|\Delta x| + |\Delta y| \leq \text{DELTA}(L)$ L is incremented by 1. The parameter L also keeps track of which regions are to be used on the next iteration. $\text{DELTA}(3)$ is the final threshold. Once the computed correction is reduced below this threshold the program assumes that the images are aligned and returns control to the supervisor. For further detail, refer to the flow diagram Figure 10.

The search scheme developed in this section has been shown to satisfy all four criteria listed above. In addition, it makes efficient use of computation time and has worked well during tests. For further detail, refer to Figure 7.

C. Angular Alignment

Angular alignment occurs at three points in the algorithm and software: 1) coarse angular alignment, 2) fine angular alignment and 3) a fine angular correction or "dithering" after each least squares estimate of Δx , Δy . In all three cases the measure used to determine alignment is the cross-correlation of the magnitudes of the Fourier transforms of the reference and misaligned image sampled at the frequencies chosen during spatial frequency selection. Let $m_{r,i}$ be the magnitude of the reference transform at spatial frequency i and let $m_{a,j}$ be the magnitude of the misaligned transform at spatial frequency j . Then the Cauchy-Schwarz inequality² states that

$$\left(\sum_{i=1}^n m_{r,i} m_{a,i} \right)^2 \leq \left(\sum_{i=1}^n m_{r,i}^2 \right) \left(\sum_{i=1}^n m_{a,i}^2 \right) \quad (3)$$

with equality if and only if $m_{a,i} = m_{r,i}$ for every i . Hence,

² D.L.Kreider, et al., An Introduction to Linear Analysis, Addison-Wesley, 1966, chap. 7.

the measure

$$SUM = \frac{\left(\sum_{i=1}^n m_{r,i} m_{a,i} \right)^2}{\sum_{i=1}^n m_{a,i}^2} \quad (4)$$

achieves its maximum when $m_{a,i} = m_{r,i}$ for every i . From equation (1) this condition will occur when the two images are aligned in angle. Angular alignment is achieved by trying various angles, computing SUM and picking the angle where SUM is maximum. Three different search schemes are used at the three sections in the algorithm where there is an angular alignment.

The first alignment is coarse angular alignment. The misaligned image is assumed to be misaligned in angle within some maximum displacement. (In the software this displacement is ± 6 degrees and is stored in the variable CORSE.) During coarse angular alignment the light table is first rotated cw 6 degrees and SUM computed. The table is then rotated ccw in 2 degree steps over a 12 degree sector. At each step SUM is computed and compared with the previous largest value of SUM which is stored in the variable MAX. If $SUM > MAX$ then MAX is replaced by SUM and the angular position saved in the variable STEPS. After the table has been rotated over the 12 degree sector it is rotated back to the position of maximum SUM. This ends the coarse angular alignment. For more detail, refer to the program flow diagram Fig. 8.

The next phase is fine angular alignment. At the beginning of this phase the light table is at the position of maximum SUM determined during coarse alignment. The variable MAX holds this value of SUM. The angle step size is set initially to a value stored in the variable FINE. (This value is 1.6 degrees in the program.) Current step size is stored in the variable STEPS. The light table is then rotated cw STEPS degrees and SUM is computed. If $SUM > MAX$ a flag is set and SUM replaces MAX. The light table is then rotated ccw $2 \times STEPS$ degrees and SUM is computed. If $SUM > MAX$ a flag is set and SUM replaces MAX. The light table is then rotated to the position where SUM was maximum (either where

it was initially or \pm STEPS degrees from that position.) Then STEPS is divided by 2. The above three-position search is repeated iteratively until STEPS has been reduced below 0.1 degree. This one-dimensional search technique converges quickly to the maximum SUM to within 0.1 degrees. Since only two evaluations of SUM are required per iteration, a total of 10 evaluations are required during fine angular alignment. By comparison, a brute force search over 4 degrees in 0.1 degree steps would require 40 evaluations of SUM. Other search techniques such as Fibonacci and golden section³ were investigated during the development of the algorithm. However, because they maximize a function of a continuous variable they were not applicable since angle increments are discrete, 0.1 degree steps in the image alignment system. Once STEPS has been reduced below 0.1 degree fine angular alignment is complete. For more detail, refer to program flow diagram Figure 9.

As discussed in the Overview, since each sensor does not see the exact same image because of misalignment, equations (1) and (2) are only approximate during alignment. The coarse and fine angular alignment must occur before estimation of Δx and Δy . This is because equation (2) requires that $\Delta\theta = 0$. However, since $\Delta x \neq 0$, $\Delta y \neq 0$ during angle alignment, equation (1) is only approximate and very likely the position of maximum SUM will not correspond exactly with $\Delta\theta = 0$. Hence, a residual angle error will generally exist after fine angular alignment. This residual error will prevent Δx and Δy from being estimated exactly since equation (2) will only approximately hold. However, if Δx and Δy can at least be reduced then it should be possible to further reduce $\Delta\theta$. This will, in turn, allow better estimates of Δx , Δy . Hence, the algorithm was designed to be iterative. That is, after fine angle search a series of iterations consisting of a least squares estimation followed by a fine angle correction is carried out. This approach has worked well in practice. Depending on the test image, $\Delta\theta$ can be as large or larger than 1.0 degree after fine angular alignment. However, as Δx and Δy are reduced $\Delta\theta$ will

³ D. J. Luenberger, Introduction to Linear and Nonlinear Programming, Addison-Wesley, 1973, chap. 7.

be reduced to a few tenths of a degree.

Fine angle correction is very similar to a single iteration during fine angular alignment. After Δx and Δy have been estimated and the light table moved to eliminate these estimated errors, SUM is recomputed and stored in MAX. The light table is then rotated cw 0.1 degree and SUM is computed. If $SUM > MAX$ a flag is set and MAX is replaced with SUM. The light table is then rotated ccw 0.2 degrees and SUM is computed. If $SUM > MAX$ a flag is set and MAX is replaced with SUM. The light table is then rotated to the position where SUM was maximum. Hence, up to ± 0.1 degree of correction can be applied each iteration. For further detail refer to flow diagram Figure 10. Both fine angular alignment and fine angular correction is accomplished in software in subroutine FINSCH.

D. Least Squares Estimation of ΔX and ΔY

Estimation of Δx and Δy is based on equation (2). This equation states that if $\Delta\theta = 0$ then the difference in phase between the reference and misaligned transform is a bilinear function of Δx and Δy . By measuring this phase difference at the previously chosen set of spatial frequencies Δx and Δy can be estimated by fitting a plane to the data. A standard means of curve fitting is least squares estimation².

To apply least squares estimation, the first step is to sample the transforms at the pre-selected spatial frequencies using the DEFT sensors and compute the phase from the real and imaginary parts of the transform. This rectangular-to-polar conversion is accomplished using the Cordic algorithm⁴. Both the transform magnitude and phase are computed. (In the software, this is implemented in subroutine CORDIC.) The transform phase as provided by the DEFT sensor is of the form $\phi_R(\omega_x, \omega_y) = \phi_{TR}(\omega_x, \omega_y) + \phi_E(\omega_x, \omega_y) + \phi_{RO}$ (5) for the reference image and

$$\phi_A(\omega_x, \omega_y) = \phi_{TA}(\omega_x, \omega_y) + \phi_E(\omega_x, \omega_y) + \phi_{AO} \quad (6)$$

for the misaligned image where ϕ_R and ϕ_A are the total phases of the reference and misaligned images, respectively. ϕ_{TA} is the phase due

² op. cit.

⁴ J.E. Volder, "The Cordic Trigonometric Computing Technique," IRE Trans. Comp. Sept. 1959, pp. 330-334.

to the Fourier transform of the reference image. ϕ_{TA} is the phase due to the Fourier transform of the misaligned image. ϕ_E is phase due to the electronic detectors, filters and cables. ϕ_{RO} and ϕ_{AO} are constant phase terms associated with the reference and aligning sensor. $\phi_{RO} \neq \phi_{AO}$. This phase difference arises from the non-equal length of cables connecting the sensor modules to the computer cabinet as well as from sensor mismatch such as acoustic wave velocity differences. ϕ_E is approximately equal for both modules. Hence, if the images are perfectly aligned, $\phi_R - \phi_A = \phi_{RO} - \phi_{AO}$. Hence the term $\phi_{RO} - \phi_{AO}$ must be subtracted from the left side of equation (2) since it arises from the sensors and electronics and has nothing to do with the Fourier transform. The first step in least squares estimation then is to measure $\phi_{RO} - \phi_{AO}$. Recall that for any image which is an intensity function

$$\phi_{TR}(0,0) = \phi_{TA}(0,0) = 0 \quad (7)$$

Hence

$$\phi_A(0,0) - \phi_T(0,0) = \phi_{RO} - \phi_{AO} \quad (8)$$

Hence, $\phi_{RO} - \phi_{AO}$ can be measured by sampling the two transforms at zero spatial frequency, computing the phase and subtracting the respective phases. Since the transform always has maximum magnitude at zero spatial frequency, the signal-to-noise ratio for this measurement will be good. The computed phase difference $\phi_{RO} - \phi_{AO}$ must be adjusted to lie in the interval $(\pi, -\pi]$. In the software, this adjustment is made in subroutine PHASDF. Subroutine PHSSET measures the phase difference and stores the result in the variable PHASE.

Once PHASE has been computed, the left side of equation (2) can be evaluated. At each of the selected spatial frequencies ω_{xi}, ω_{yi} the two transforms are sampled and the phase computed. The difference

$$P_i = \phi_R(\omega_{xi}, \omega_{yi}) - \phi_A(\omega_{xi}, \omega_{yi}) - \text{PHASE} \quad (9)$$

is formed and adjusted to lie in the interval $(\pi, -\pi]$. During any given iteration, only those spatial frequencies lying in the regions whose indices are less than or equal to L are

evaluated. Then the following quantities are computed:

$$UU = (1/K\$G) \sum_{i=1}^L S1U(i) \quad (10)$$

$$VV = (1/K\$G) \sum_{i=1}^L S1V(i) \quad (11)$$

$$UV = (1/K\$G) \sum_{i=1}^L S1UV(i) \quad (12)$$

$$UP = (1/KKKK) \sum_{i=1}^L S1UP(i) \quad (13)$$

$$VP = (1/KKKK) \sum_{i=1}^L S1VP(i) \quad (14)$$

The variables UU through VP are computed in subroutine SUMPAR from partial products S1U through S1VP which are computed in subroutine MEASRE. K\\$G and KKKK are scale factors which scale the corrections Δx and Δy . The variables S1U through S1VP are defined as

$$S1U(i) = \sum_{j=k}^l (cf_{xj})^2 \quad (15)$$

$$S1V(i) = \sum_{j=k}^l (cf_{yj})^2 \quad (16)$$

$$S1UV(i) = \sum_{j=k}^l c^2 f_{xj} f_{yj} \quad (17)$$

$$S1UP(i) = \sum_{j=k}^l cf_{xj} p_j \quad (18)$$

$$S1VP(i) = \sum_{j=k}^l cf_{yj} p_j \quad (19)$$

where $f_{xj} = \omega_{xj}/2\pi$, $f_{yj} = \omega_{yj}/2\pi$ and c is a constant used to scale the fixed point frequency variables used by the program. The limits k and l are defined by

$$k = BDRYCT(i-1) + 1 \quad (20)$$

$$l = BDRYCT(i) \quad (21)$$

BDRYCT is a table in software and the entry BDRYCT(i) is the number of spatial frequencies in region 1 through region i. BDRYCT(0) is defined to be zero. If $k > l$ then there are no

points in region i and the S_1 terms are defined to be zero for that i . Hence, the indices k and ℓ limit the partial products S_1 to contain contributions from only those spatial frequencies within region i . The terms UU through VP contain contributions from spatial frequencies in regions 1 through region L .

With these definitions the least squares solution for Δx and Δy is given by²

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \frac{1}{K} \begin{bmatrix} UU & UV \\ UV & VV \end{bmatrix}^{-1} \begin{bmatrix} UP \\ VP \end{bmatrix} \quad (22)$$

The gain constant $1/K$ equals 1 during the first 12 iterations. After 12 iterations it equals $1/(i-11)$ on the i th iteration. Hence, after 12 iterations, the computed correction is weighted by a gain factor which decreases each succeeding iteration. This has the effect of making the final alignment insensitive to any noise in the phase differences P_i . This technique is a variation of the Robbins-Monro procedure for finding the root of a function in the presence of noise. A readable discussion of this procedure is contained in⁵. The harmonic sequence of weights $\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots\}$ has the property that

$$\lim_{n \rightarrow \infty} \frac{1}{n} = 0 \quad (23)$$

while

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty \quad (24)$$

That is, the computed corrections will always be reduced to zero while the total corrective effort is potentially unlimited. It is shown in⁵ that random experimental errors will be cancelled out using this sequence of weights. However, another important reason for using this sequence was to insure that the program will satisfy its stopping condition after a reasonable number of iterations. This stopping condition is that

$$| \Delta x | + | \Delta y | \leq .04 \text{mm} \quad (25)$$

It was found through experimentation that reducing the right hand side of this equation did not improve the accuracy of

⁵D.J.Wilde, Optimum Seeking Methods, Prentice-Hall Inc. 1964.

algorithm but did increase its running time.

For additional detail refer to flow diagram Fig. 10. In this figure the notation (KKKK + 2) means a variable stored in RAM two bytes after the address of the label KKKK. The corrections Δx and Δy are computed iteratively. On the first iteration $L = 1$. L is incremented by one, if any one of the following conditions holds:

$$1) |\Delta x| + |\Delta y| \leq \text{DELTA} \quad (26)$$

$$2) |\Delta x| + |\Delta y| > 2\sqrt{2} \quad (27)$$

$$3) \text{BDRYCT}(L) < 2 \quad (28)$$

Condition 1) means that a larger region is used as soon as the correction is small enough so that there can be no phase ambiguity. $\text{DELTA} (3) = .04\text{mm}$ and is the stopping condition since NBDRY is set to 3. Condition 2) means that the computed correction is too large. The correction is skipped and a larger region is used on the next iteration. Condition 3) means that in the present region there is at most one spatial frequency. Since at least two spatial frequencies are required for the least squares estimation of Δx and Δy , the region is enlarged in hopes of increasing the number of spatial frequencies on the next iteration.

The partial products $S1U$, $S1V$ and $S1UV$ depend only on the chosen spatial frequencies. Hence, they are only computed once during spatial frequency selection. The partial products $S1UP$ and $S1VP$ depend on both the spatial frequencies and the transform phase at those frequencies. Hence, they must be recomputed each iteration.

During normal operation using both sensors, fresh magnitude and phase data from both sensors is collected each iteration. In addition, the origin phase difference is re-measured each iteration. Hence, any zero mean noise in these measurements will be averaged out over a number of iterations.

E. Alignment Algorithm: Alternative Modes of Operation

To add flexibility to this experimental alignment system the basic algorithm and software implementation has been augmented with two alternative modes of operation.

In order to test the alignment capabilities of the system for test images which are misaligned in translation only, the software can function in a mode which skips all angular alignment steps. That is, coarse angular alignment, fine angular alignment and fine angular correction (or dithering) are all skipped. In the program, the flag HOW is set to zero if these functions are to be skipped and set to one if they are to be executed. Refer to flow-diagrams Figures 6 through 10 for details. Angular alignment can be skipped during both the normal two-sensor mode of operation and also during the single sensor mode which will be described next.

To this point the description of the alignment algorithm has dealt exclusively with the normal two-sensor operation. In this mode, one sensor views the fixed light table which contains the reference image. The second sensor views the movable light table which contains the misaligned image. In the single sensor mode of operation, only the sensor which views the movable light table is used. The image on this table functions as both the reference and as the misaligned image. This is accomplished as follows: During spatial frequency selection the movable image is viewed and spatial frequencies chosen from its transform. Both the magnitude and phase of the transform at these frequencies are then sampled and stored in a table. Magnitude samples are stored in table MS while phase samples are stored in table PS. These samples will not be updated until a new reference alignment is initiated. The alignment program then returns control to the supervisor so that the user can move the light table under manual control. If the alignment program is re-entered, it will seek to re-align the misaligned image to its position during reference alignment

using the tables MS and PS as a reference.

The structure of the alignment algorithm is identical for both two-sensor and one-sensor operation with the following exceptions:

1) The flag SNFLAG = 0 for two-sensor operation and SNFLAG = 1 for one-sensor operation.

2) Since the two DEFT sensors have slightly different origin frequencies, near the beginning of single sensor reference alignment the reference sensor origin frequency variables X0ZERO, Y0ZERO are replaced with the origin frequency values contained in the aligning sensor frequency variables X1ZERO, Y1ZERO.

The subroutine RDDEFT is used to sample the DEFT sensors and average either one or sixteen samples at each spatial frequency. (Single samples are taken only during the initial search for prominent spatial frequencies.) When the flag SNFLAG = 1, RDDEFT will always sample from the sensor which views the movable table (aligning sensor). By also replacing the origin frequencies the program operates normally but only receives data from the aligning sensor.

3) Since only a single sensor is used, no sensor phase mismatch exists. Hence, when subroutine PHSSET is entered the condition SNFLAG = 1 causes an immediate return to the main alignment program and no phase difference is required.

4) During the least squares estimation iterations, new reference phase samples cannot be taken since the reference orientation no longer exists. The flag SNFLAG causes this step to be skipped. (See the flow diagram for subroutine MEASRE, Figure 11. Phase samples are normally taken during MODE = 3.)

After a sequence of one-sensor alignments, the table of variables used in the alignment program must be re-initialized by pressing F followed by E. If this is not done and a correlation is attempted, then SNFLAG = 1 and only one sensor will be used erroneously. (That is, the movable image will be correlated against itself.) If instead, a two-sensor alignment is initiated, then SNFLAG will be reset to 0. However X0ZERO and Y0ZERO will

not be restored to their proper values. This will lead to larger than desired errors in alignment.

F. Algorithm Operation Count

In this section a simplified operation count for the Fourier transform-based alignment algorithm will be developed and compared with the operation count of an algorithm which aligns two images using the image intensity functions. For this development an operation is defined to be either a multiplication or a division.

Consider first the Fourier transform-based algorithm.

Let n_s be the number of spatial frequencies used in the algorithm. The cross-correlation of two sequences of length n_s requires $2n_s + 2$ ops. An additional $2n_s$ operations are required to form the magnitude from the real and imaginary components of the transform. (The Cordic algorithm takes about as long as a multiply.) Let n_θ be the number of correlations required for an alignment. (n_θ is typically 20 - 30.) The angle correlation requires $n_\theta(4n_s + 2)$ ops. One iteration of least squares estimation requires $7n_s + 12$ ops. This count includes Cordic, formation of UU through VP and matrix inversion.

Let n_{ls} be the number of iterations of least squares. (Typically, n_{ls} is less than 10.) An additional count of 2200 ops are required during the search for prominent spatial frequencies. (These are all calls to CORDIC.) The total number of operations is

$$n_\theta(4n_s + 2) + n_{ls}(7n_s + 12) + 2200 \quad (29)$$

For the typical values $n_\theta = 30$, $n_x = 16$, $n_{ls} = 10$ the first term contributes 1980 ops, the second term contributes 1240 ops and the total is 5420 ops.

Consider now the intensity function-based algorithm. This algorithm works as follows: A three-dimensional grid of possible Δx , Δy , and $\Delta \theta$ values is searched. At each point in the grid the misaligned image is translated and rotated to the corresponding grid positions and the two images are correlated.

It may not be necessary to correlate every pixel. Let n_p be the number of pixels used. Then a single correlation requires $2n_p + 2$ ops. Let n_x , n_y and n_θ be the number of increments of Δx , Δy and $\Delta \theta$ to be tested. Then the number of operations for alignment is

$$n_x n_y n_\theta (2n_p + 2) \text{ ops} \quad (30)$$

Additional operation would be required to choose the n_p pixels.

For simplicity these will be ignored. To arrive at a number for comparison, assume that this algorithm searches a

5 mm x 5 mm x 12 degree cube with grid spacing .05 mm in x and y and .1 degree in θ . Then, $n_x = n_y = 100$ and $n_\theta = 120$.

Assume $n_p = 16$. Then 40.8×10^6 ops are required for alignment. There are a number of multidimensional search techniques which will reduce this count considerably. However, in all circumstances the Fourier transform-based algorithm is more efficient since a multidimensional search is not required.

IV. SYSTEM SOFTWARE

A. General Description

The Image Alignment and Correlation System has been programmed to perform fourteen separate functions. The control programs for these functions are stored permanently in UV-erasable read-only memories at addresses $A000_H$ - $BFFF_H$. (All address references are given here in hexadecimal notation.) The software is organized as a supervisor or main program and a master subroutine for each of the three major system functions, which are display generation, image alignment, and image cross-correlation. The master subroutines use a number of smaller subroutines, many of which are shared.

In addition to the 8K of software written in the performance of this contract, the system includes a library of mathematical functions which resides at 8000_H - $83FF_H$ and a slightly modified version of the FANTOM-II monitor, located at $FC00_H$ - $FFFF_H$. Both of these items were purchased from Wintek Corp.

The system software uses RAM at 1700_H - $1FFF_H$ and $EC00_H$ - $EDFF_H$ for temporary storage of data and parameters. The remainder of RAM is available for future expansion or software experiments.

The following sections will describe the supervisor and the three master subroutines. The smaller subroutines, except for those which are self-explanatory from their listings, are described in Appendix A. Complete program listings for all of the Deft-written software, as generated by the Wintek assembler, are found in Appendix B. The math library and the FANTOM-II monitor are documented with the Commercial Data.

B. Supervisor Program ($A91C_H$ - $AA5D_H$)

When power to the system is turned on, a delay circuit on the auxiliary card (next to the DAS card) holds the microprocessor subsystem's reset line near ground until the power supply voltages

and the clock frequency have stabilized. The small button on the auxiliary card also grounds the reset line. When the reset line goes high, the processor reads the contents of memory at hex addresses $FFFE_H$ and $FFFF_H$, and loads them into the program counter. Execution begins from that point.

As supplied by Wintek the reset vector is $FE07_H$, which is the reset entry point of the FANTOM-II monitor. For the Image Alignment and Correlation System the reset vector in the FANTOM-II EPROM was changed to $A91C_H$, and $FE07_H$ was used instead as the vector for non-maskable interrupts (NMI) which is stored at $FFFC_H$ - $FFFD_H$. This change allows the other small button behind the door, which grounds the NMI line, to stop execution and transfer control to the FANTOM-II monitor. Except for these four bytes, the monitor in the system is identical to FANTOM-II as documented with the Commercial Data.

When execution begins at $A91C_H$ following a system reset, the first instructions set up the peripheral interfaces and initialize certain parameters in read/write, or random access memory (RAM). The supervisor then enters a wait loop at $A9B9_H$ with dashes displayed on the front-panel LED readout. The dashes indicated that the system is ready to accept commands from the keypad directly below the readout.

Pressing a key at this point will result in the display of the corresponding program number, with two exceptions. They are "A", which is reserved for ABORT, and "E", which is used for EXECUTE. After entering a valid program number, pressing "E" will cause execution of that program to begin. Any time before "E" is pressed, entering a new program number will override the preceding entry. Pressing "E" initially will have no effect. Entering "A" at any time will cause a return to the supervisor, and the execution of any program in progress will be terminated. Decoding of the "A" key and updating of the LED display is done in the interrupt service routine.

Near the end of the wait loop a test is made (at $AA07_H$)

to see if the system is in a calibrated state for alignment. If so, another test is made to sense whether any of the stepper motors is being actuated manually. If the clock pulse which moves a motor is detected, further tests are done to identify the motor. "Error" is displayed if the motor cannot be identified, and if this should occur it would indicate a hardware failure. After the stage in motion is identified a subroutine (POSDIS) is called which updates the counters which keep track of the positions of the translation and rotation stages when the system is calibrated. This subroutine also displays these positions. On returning from the subroutine the program checks for the end of the stepper motor pulse so that one pulse is not counted as two.

The entry addresses of the master subroutines are stored in a table at A800_H. When the "E" key is pressed with a valid program number in place, the program number is used to point to the corresponding entry address, and a jump-to-subroutine (JSR) at that address is executed. Upon returning, the program number is examined and the display is either left unaltered or cleared and filled again with dashes.

C. Plotter and CRT Display Routine (ABC0_H - ADE2_H)

Entry to this program at AC47_H first initializes a number of parameters and clears a 256-byte area in RAM where the largest current vertical deflection values will be stored. It then tests the program number to determine which image is to be displayed, and sets the analog multiplexer in the DAS accordingly. A heading is written on the CRT display, showing the starting points for the frequency scan and identifying the image whose transform is being displayed.

The action of the program from this point depends on whether the x-y plotter or the CRT display is being executed, and this distinction is coded in the least-significant bit (LSB) of the program number. In the case of the plotter the pen is retraced to the lower left corner and dropped to the writing position. In

the case of the CRT display, which is vector driven, a dark vector is written to the lower left corner. The synchronous detector subroutine then samples the analog outputs from the appropriate synchronous detector circuit card and converts them to a complex digital representation of the sensor output. The detector subroutine returns the sums of sixteen consecutive samples for both the real and imaginary parts.

At this point the x-axis synthesizer frequency is increased by 40 kHz to allow the subsequent processing time for frequency stabilization. An extra time delay is inserted if the synthesizer crosses the boundary of 40 MHz, but this will not occur unless the starting frequencies or increments are changed. The absolute magnitude of the complex signal value is then computed by the CORDIC subroutine and a scaling or gain adjustment is made by shifting the result to the left.

The signal magnitude then is added to the raster height and compared to the previous value at the same horizontal position. For the first line of the raster the "previous" values are all zero. If the new value exceeds the old value, the new value is stored and either the pen is moved accordingly or a vector is drawn on the CRT display. If the old value is greater, the pen is lifted, or a character denoting a dark vector is sent to the CRT display, so that raster lines behind peaks are hidden.

This sequence is repeated 192 times for each line of the raster. At the end of each line the y-axis synthesizer frequency is incremented by 120 kHz and the x-axis frequency is returned to its starting point. The position of the raster line is moved up by a count of 2 for the CRT display or 3 for the plotter. The number of the raster line becomes the initial x-coordinate to tilt the raster, and either the pen is lifted and retraced to that point or a dark vector is written to retrace the CRT display.

Sixty-four lines are drawn in this manner. At the end of

the last line the pen is lifted and moved to the lower right corner if the x-y plotter is being driven, so that the paper can be removed easily. Then a bell code is sent to the terminal, followed by a return to the supervisor.

D. Alignment Program and Its Subroutine

The alignment program is written in the form of a main subroutine which is called by the supervisor program under control of the matrix key pad. Additional subroutines are called by the alignment program. The logical flow of the program is complex and is best understood by study of the assembly language listing in Appendix B. Flow diagrams are provided here which show the functional flow of the program. These diagrams can be used to relate the algorithm to its assembly language implementation.

Refer to the overall flow diagram, Figure 6. More detailed flow diagrams are shown in subsequent figures. The alignment program operation is controlled by three variables SNFLAG, HOW and CALIBR. These variables are set either before entry or upon entry to the alignment program and remain constant during each call to that program. For the various modes of operation, their values are indicated in Table 4. As shown, CALIBR indicates whether an alignment is to be a reference or a subsequent alignment. SNFLAG indicates whether one sensor or two sensors are to be used. HOW indicates whether the alignment is to include rotation.

The functional flow of the alignment program closely follows the description of the algorithm given in Section III.

The main subroutine which is called by the alignment program is MEASRE. This subroutine has a number of functions controlled by variables MODE, SNFLAG, SFLAG and ADRSW. The flow diagram of MEASRE is shown in Figure 11. When MODE = 1, MEASRE is used in spatial frequency selection. When MODE + 2 MEASRE is used in angular alignment. When MODE = 3 MEASRE is

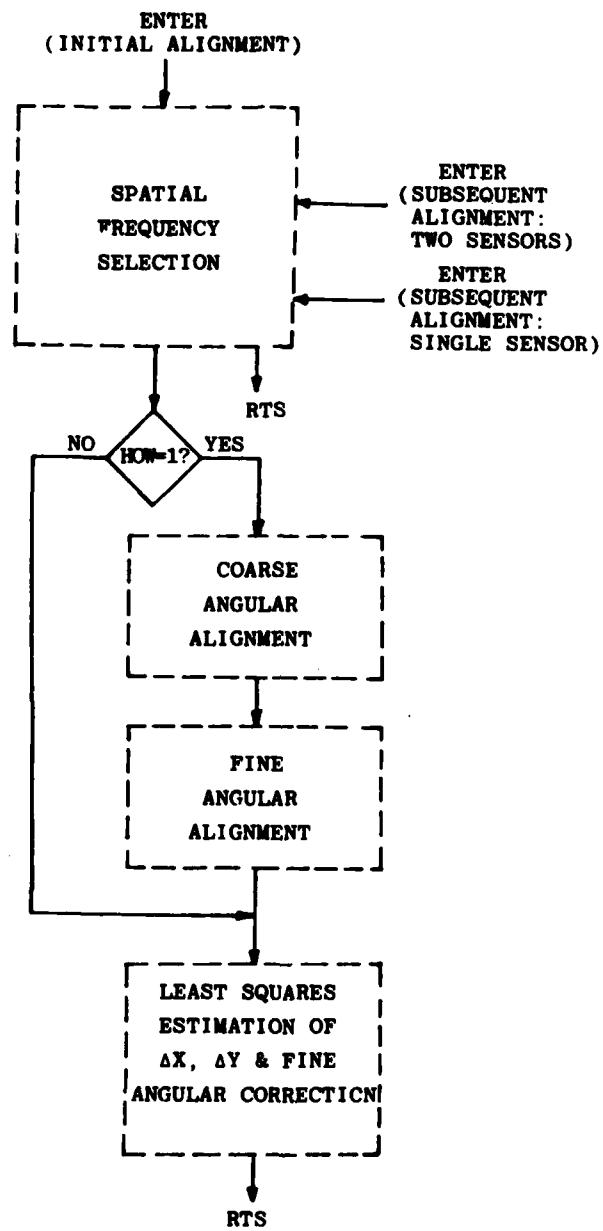


Figure 6 - Overall Flow Diagram for Alignment Program

TABLE 4

Modes of Operation	Flags		
	CALIBR	SNFLAG	HOW
Initial alignment/two sensors	0	0	1
Initial alignment/one sensor	0	1	1
Subsequent alignment/two sensors/ rotation	1	0	1
Subsequent alignment/two sensors/ no rotation	1	0	0
Subsequent alignment/one sensor/ rotation	1	1	1
Subsequent alignment/one sensor/ no rotation	1	1	0

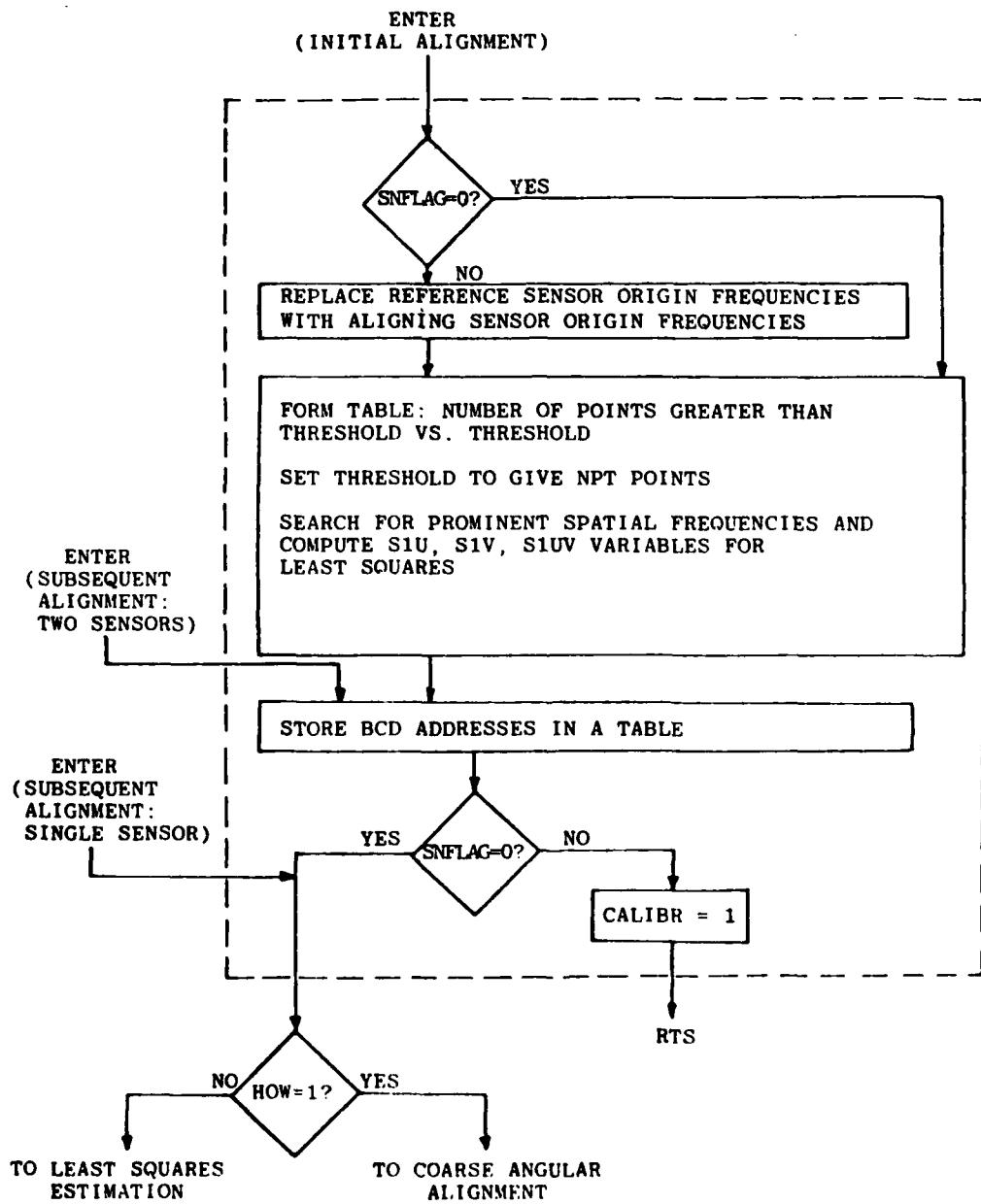


Figure 7 - Flow Diagram for Spatial Frequency Selection

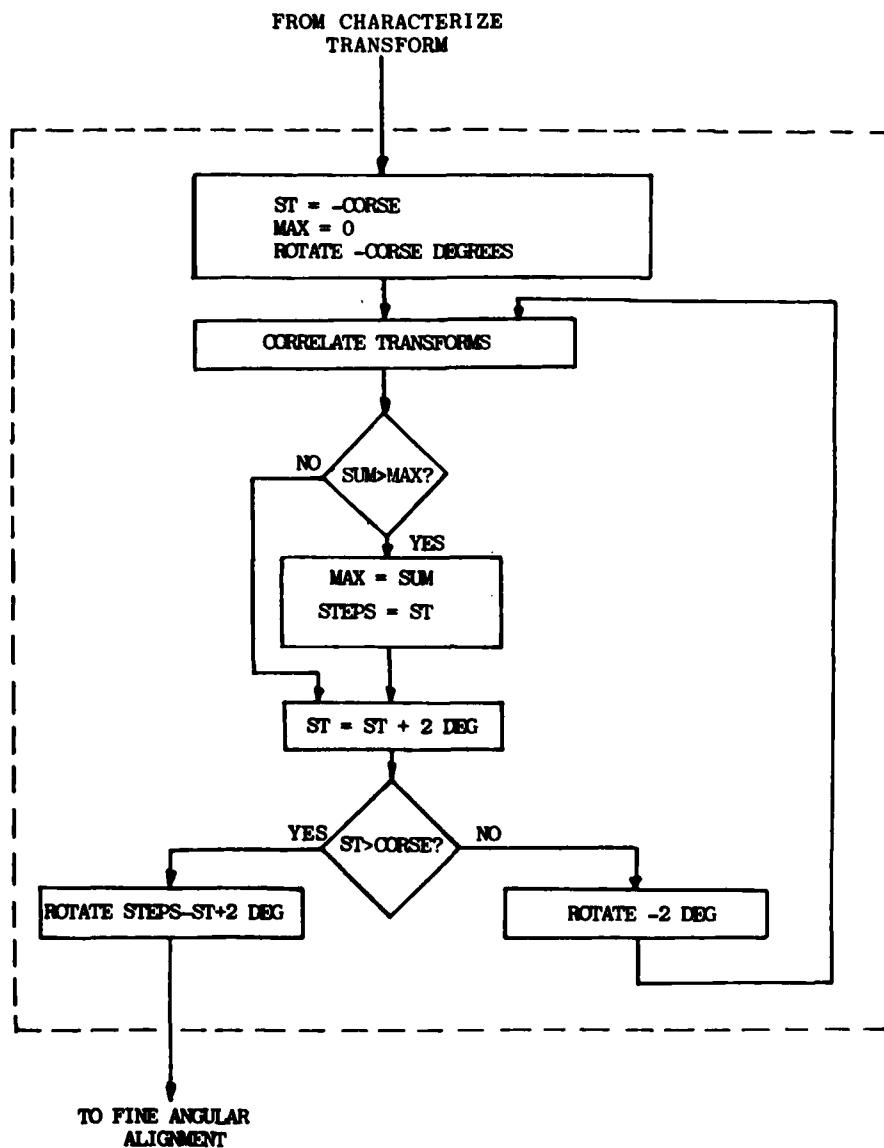


Figure 8 - Flow Diagram for Coarse Angular Alignment

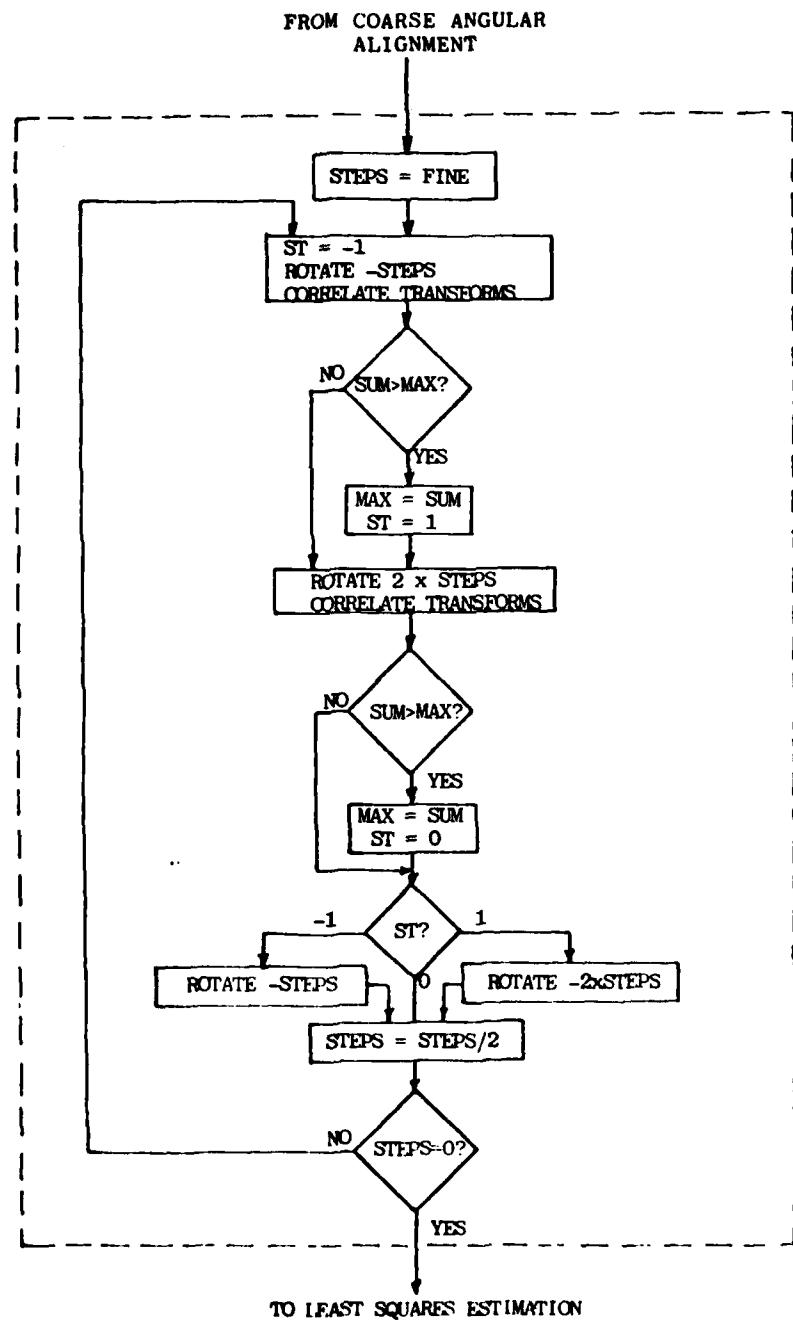


Figure 9 - Flow Diagram for Fine Angular Alignment

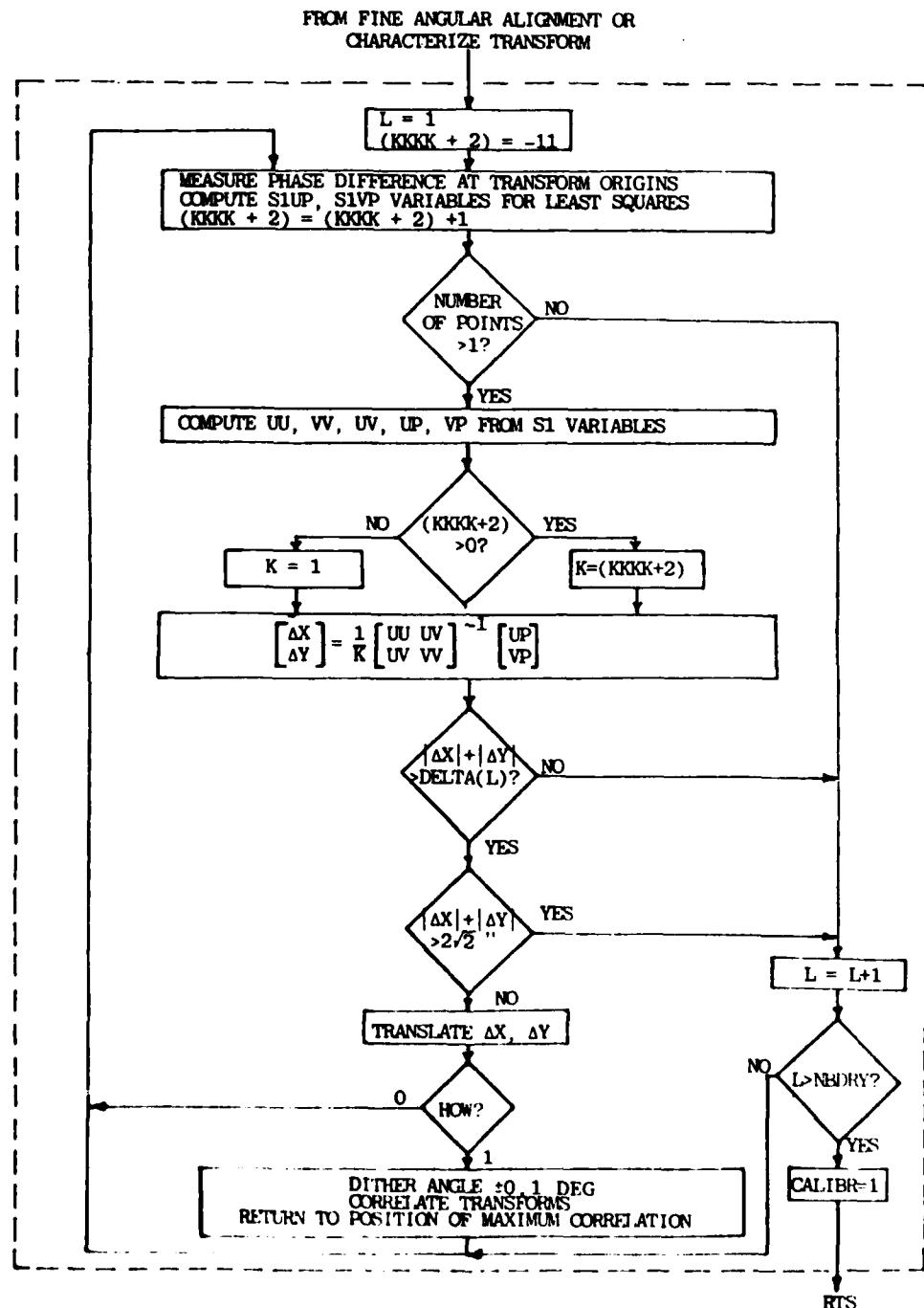


Figure 10 - Flow Diagram for Least Squares Estimation of $\Delta X, \Delta Y$ and Fine Angular Correction

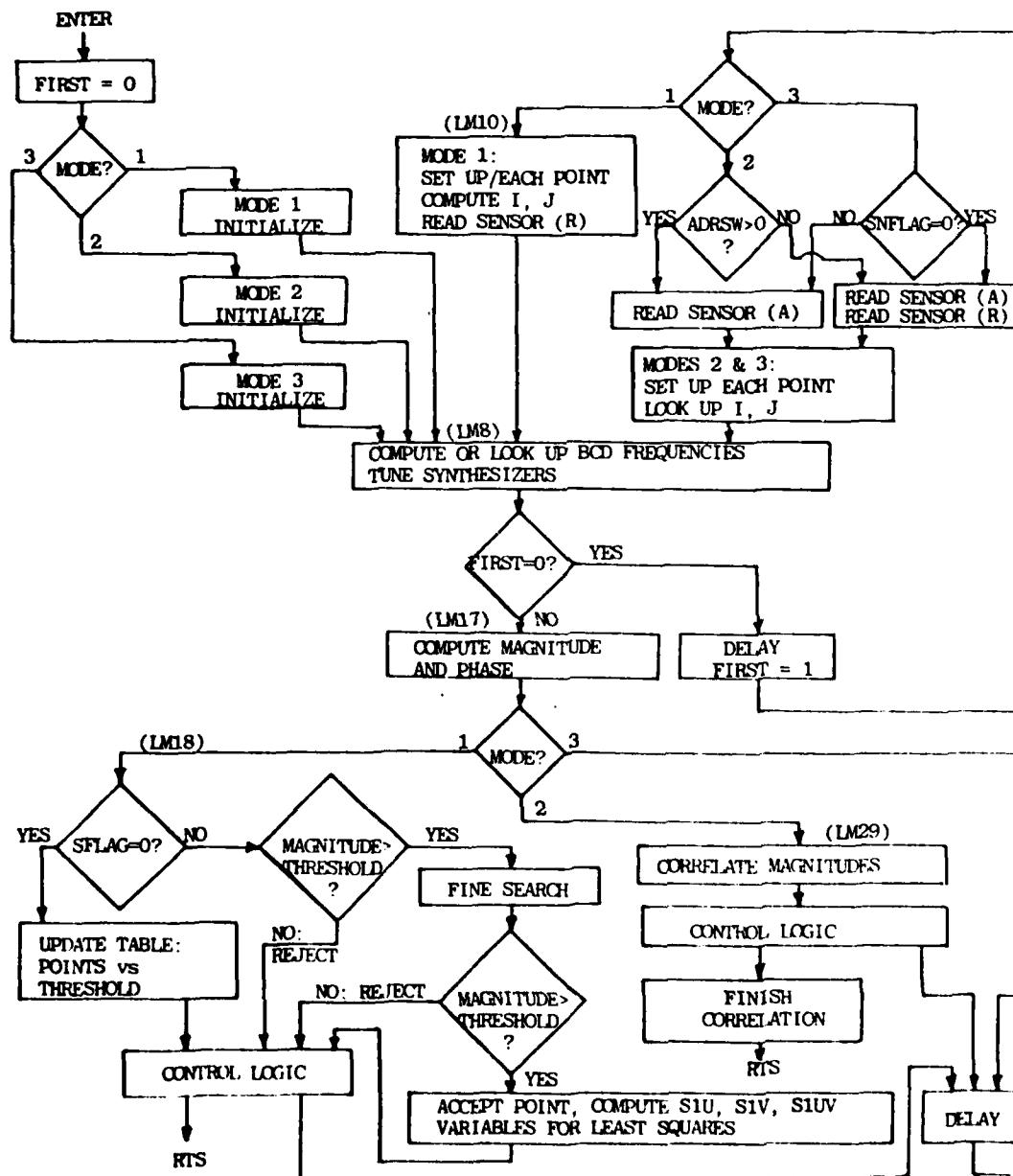
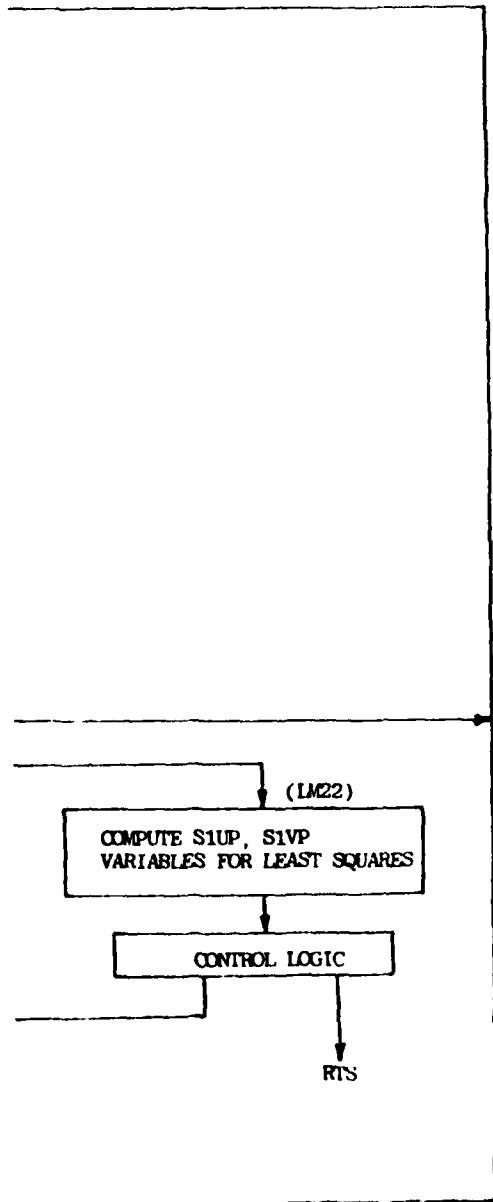


Figure 11 - Flow Diagram for Subroutine MEASRE



is used in least squares estimation.

Consider first MODE = 1 operation. MEASRE is called twice. During the first call SFLAG = 0. The transform is searched and a table characterizing the transform is set up using subroutine UPDATE. (See section on alignment algorithm for a discussion.) During the second Call SFLAG = 1. The transform is again searched and the prominent spatial frequencies are selected. In addition, during this second call the variables S1U(i), S1V(i) and S1UV(i) are computed and stored since their values depend only on the spatial frequencies and do not change until there is another initial alignment.

Consider now MODE = 2 operation. During the first call to MEASRE, ADRSW = -1. This causes the frequency addresses of the prominent spatial frequencies to be computed and stored in Table BCDAR. In addition, the magnitudes of the reference transform at these frequencies are stored in Table MS.

Subsequent calls to MEASRE are used to compute SUM as defined by equation 4. Hence, MEASRE computes the function "correlate transforms" shown in Figures 8, 9 and 10. In all cases ADRSW = 1 so that fresh data from the misaligned sensor will be taken and correlated against reference data which has been stored in the Table MS.

Finally consider MODE = 3 operation. The variables S1UP(i) and S1VP(i) are computed for least squares estimation. If SNFLAG = 0 then fresh data from both sensors is used to compute these variables. If, instead, SNFLAG = 1 then fresh data from the misaligned sensor is used along with stored phase data from Table PS to compute these variables.

The remainder of this section consists of a brief description of the major subroutines and variables used by the alignment program. The notation (VAR1, VAR2) is used to indicate a two-byte variable consisting of the one-byte word VAR1 which is followed in memory immediately by the one-byte word VAR2.

Subroutine FINSCH is called during fine angular alignment and fine angular correction (dither angle). It computes SUM at the present position and at \pm FINE degrees from it. It then moves the light table to the position of maximum correlation.

Subroutine SUMPAR computes UU, UV, VV, UP and VP from the partial products S1U, S1V, S1UV, S1UP and S1VP. The variable L is used to set the summation limits in the defining equations (10) through (14).

Subroutine CORDIC computes the Cordic algorithm with 16 iterations. The algorithm has two functions. First, if variables COR9 = COR10 = 0 upon entry then the two byte rectangular coordinates (COR1, COR2), (COR3, COR4) will be converted to polar form with (COR1, COR2) = $.41169 \times$ magnitude and (COR9, COR10) = phase (radians/ π). If the two-byte variable (COR9, COR10) \neq 0 then the rectangular coordinates (COR1, COR2), (COR3, COR4) are rotated through the angle (COR9, COR10) and suffer a gain change of .41169.

Subroutines ADDRES computes the BCD frequency variables UI1, UI2, VJ1, VJ2 from the binary frequency variables I, J. In addition, under control of variable ADRSW the BCD variables can be either stored in Table BCDAR after computation or read from that table in lieu of computation.

Subroutine READ is used to read data from the reference sensor and store the magnitude and phase in MP and PS respectively.

Subroutine BI\$BCD computes a BCD number from a binary number.

The PUSH and PULL subroutines are used to push or pull four-byte numbers on or off the stack for use with the MATH chip.

The MATH chip contains software which implements fixed point and floating point arithmetic operations. The use of this software is described in the manual from Wintek Corp. supplied in the Commercial Data.

Subroutine RDDEFT takes a number of samples from one of the Deft sensors, averages them and returns the average value in (real) (COR1, COR2) and (imag.) (COR3, COR4). If SENSOR = -1 then NS samples of the reference transform are averaged. If SENSOR = 0 then one sample of the reference transform is taken. If SENSOR = 1 then NS samples of the misaligned transform are averaged.

Subroutine INDEX1 computes the next coarse grid point from the previous grid point during transform search.

Subroutine GSRCH performs a fine grid search around a coarse grid point. The fine grid increment is 30 kHz and an area of 180 x 180 kHz is searched.

Subroutine UPDATE updates a table during the initial transform search during spatial frequency selection. When completed, the n-th entry in this table is the number of samples for which the magnitude of the sample is greater than $n \times \text{THR2}$ where THR2 is a small constant.

Using the table constructed by UPDATE, subroutine THRSET sets the threshold THR1 so that $\text{THR1} = n \times \text{THR2}$ where the n-th entry in the table is less than or equal to NPT and the n-1st entry is greater than NPT. NPT is the desired number of spatial frequencies.

Subroutine PHASDF computes the difference of two phase samples and adjusts the difference to lie in the interval $(\pi, -\pi]$.

Subroutine PHSSET measures the phase difference at the dc peak of the two sensors and stores the result in variable PHASE. If alignment is with one sensor only then there is no phase difference. In that case, (SNFLAG = 1), the subroutine immediately returns to the calling program.

Subroutine INIZE is used to initialize a table in RAM with program constants which are stored in ROM. The area in RAM consists of the block from 1808_{H} to $1FFF_{\text{H}}$ and the block 1812_{H}

to 1891_H. Most of the memory in the first block is simply cleared to zero. These memory blocks are reinitialized whenever the key pad key F is pressed. It is then possible to modify any parameter in RAM prior to alignment by using the monitor program. To enter the monitor, open the front panel and press the right-most button. An asterisk should appear on the CRT terminal. To modify RAM, refer to the monitor reference manual supplied in the Commercial Data. After memory has been modified, press together the control key and P on the terminal. Then enter A99B and carriage return. Follow this by G, carriage return. There should now be dashes on the LED display and control has been returned to the supervisor. Caution: Memory modification should only be attempted if the user has intimate knowledge of the alignment program, assembly language and hexi-decimal notation. The program has been designed so that memory modification is not necessary during normal operation. In particular, the software adjusts automatically to the test image presented to it. That is, the program automatically characterizes the transform and determines which spatial frequencies it will use.

The remaining subroutines used in alignment are adequately explained in the assembly language listing. The remainder of this section consists of a description of the important variables used in the alignment program. Memory locations 1800_H - 1809_H hold temporary variables which will not be discussed.

PHASE holds the difference in the phase of the two transforms provided by the two Deft sensors measured at the dc peaks.

X0ZERO and Y0ZERO are the coordinates of the dc peak of the reference sensor. X1ZERO and Y1ZERO are the coordinates of the dc peak of the aligning sensor. (MHz/10)

COR1, COR2, COR3, COR4, COR9, COR10 are used to store the two-byte variables input to and output from the CORDIC subroutine. (See description of CORDIC.)

I and J hold binary numbers which address spatial frequencies. I = J = 0 addresses the dc peak of either sensor. In general, to convert from these normalized addresses to the actual electrical frequencies use the formulas:

$$f_x = 30 \text{ kHz} \times I + \text{dc peak } x\text{-coordinate} \quad (37)$$

$$f_y = 30 \text{ kHz} \times J + \text{dc peak } y\text{-coordinate} \quad (38)$$

ID and JD are the normalized frequencies of the previous grid point.

UI1 and VJ1 are the x and y frequencies of the reference sensor expressed as BCD numbers. UI2 and VJ2 are the same for the aligning sensor. They are all computed from I and J using subroutine ADDRES.

MODE controls the function of subroutine MEASRE. (See description of MEASRE.)

DTIME is a parameter which controls the delay provided by subroutine DELAY1.

JSTART is the initial value of J during the search of the transform. It is set large enough to avoid the low-frequency region of the transforms.

BDRY indicates which region of the transform the variables I and J presently address. BDRYD is the same for ID and JD.

NBDRY gives the total number of regions that the transform is divided into. This number is 3.

BDRYPT is a pointer used to index the S1 variables.

FIRST is a flag used in subroutine MEASRE to tell whether the current point is the first point.

SENSOR tells whether the reference or aligning sensor is to be used.

NSAMP is the number of samples to be averaged in subroutine RDDEFT. LOGS = \log_2 NSAMP.

BCDPTR is a pointer used to index array BCDAR.

L is the present, outermost region. It is used during least squares estimation.

LIMIT is the largest allowable value of I which limits the search area to the sensor bandwidths.

X is the number of spatial frequencies chosen by the program.

Y is used to index data stored in MS and PS.

ZERO is not used.

X1INC and Y1INC are the fine grid increments scaled by 10. Their value is 30 kHz/10.

NPT is the desired number of spatial frequencies (16).

STACK1, STACK2, PUSHST are temporary variables used by the PUSH and PULL subroutines.

(THR11, THR12) is the magnitude threshold set by the program to pass approximately NPT points.

(THR21, THR22) = 8 is a small constant used to quantize the available range of magnitude values.

HOW is a flag indicating rotation or no rotation during alignment.

IJPTR and IJPTR1 are pointers which index array IJ.

ADRSW is a flag which controls the operation of subroutine ADDRES.

SETUP is not used.

FINE = 16 is the initial fine angle increment.

K\$G is a gain constant used to scale UU, VV, and UV.

KKKK is a gain constant used to scale UP and VP.

CORSE = 60 is the angle which the light table moves prior to the coarse angular alignment.

ST is a variable used to keep track of angular position.

SAVESP is not used.

MAX is the largest correlation to date during angle alignment.

SUM is the current correlation value.

XTRAN and YTRAN hold the x and y translations for the stepper motor subroutine.

DELX and DELY are four-byte variables which are the computed translations for the stepper motor. The lower two bytes are then stored in XTRAN and YTRAN.

VP, UP, UV, and UU are the computed least squares variables which are defined in Section III.

STEPS holds the angle where maximum correlation occurred.

ANGLE holds the angle used by the stepper motor subroutine.

SMAG holds the denominator of equation (4).

SAVEZ, SAVEY, DIRECT, DTHR11, DTHR12, DTHR21 and DTHR22 are not used.

MPLX stores the current multiplexer address.

SIGNI and SIGNQ are used to hold sign bits to sign-extend the real and imaginary sample respectively.

SFLAG is a flag which indicates first or second pass through MEASRE when MODE = 1.

SNFLAG is a flag which indicates two-sensor or single-sensor alignment.

CFLAG is a flag which indicates real or complex correlation in the correlation program.

CCFLAG is a flag which indicates whether or not the scan of the transforms is complete in the correlation program.

Some of the above variables are also occasionally used for temporary storage. The alignment program also uses a block of

memory for table or array storage. These tables are listed below.

DELTA is a table holding threshold values used to determine if L should be incremented or the alignment considered complete.

BCDAR is an array which is filled with the BCD values of the spatial frequencies chosen by the program.

BDRYLF is a table which holds the boundary values of J used to determine which region of the transform a grid point is in.

BDRYCT is an array which is filled during the second call to MEASRE, MODE = 1. The i-th entry in this table is the value of the index y for the last spatial frequency chosen in region i.

S1YP, S1UP, S1UV, S1U and S1V are arrays which hold the partial products used in least squares estimation. The notation S1VP1M means that the table points to the first word in the table and the MSB of the word.

IJ is an array which holds the normalized frequency variables I and J for each spatial frequency chosen.

PS and MS are arrays which hold the phase and magnitude of the reference transform at the chosen spatial frequencies.

ARRAY is an array used in the correlation program to hold the partial products during scanning of the transforms.

E. Calculation of Image Transform Cross-Correlation Function

To compute a cross-correlation in the spatial frequency domain, first that domain is restricted to the bandwidth of the Deft sensors. The area used is a square extending from the location of the dc peak along the f_x -axis 6 MHz and along the f_y -axis ± 3 MHz. This area is quantized to a grid with spacing 100 kHz in both f_x and f_y . Let r_i be the aligning Deft sensor transform component evaluated at grid point i. Let a_i be the aligning Deft sensor transform component evaluated at grid

point i. In general r_i and a_i are complex numbers. The standard formula for cross-correlation of these samples is

$$r = \frac{\sum_{i=1}^n (r_i - m_r)(a_i - m_a)^*}{\sqrt{\sum_{i=1}^n (r_i - m_r)(r_i - m_r)^*} \sqrt{\sum_{i=1}^n (a_i - m_a)(a_i - m_a)^*}} \quad (31)$$

where

$$m_r = \frac{1}{n} \sum_{i=1}^n r_i \quad (32)$$

$$m_a = \frac{1}{n} \sum_{i=1}^n a_i \quad (33)$$

The symbol (*) means complex conjugate. Equation (31) can be found in the references defined in the context of random variables⁶ or the Cauchy-Schwarz inequality⁷.

This formula is evaluated by the correlation routine and the result displayed on the LED display and on the computer display terminal if it is connected. In general, r will be a complex number. It is displayed in polar form.

In some cases it is more desirable to compute the cross-correlation between the magnitude of the transforms. The corresponding formula is

$$r = \frac{\sum_{i=1}^n (|r_i| - m|_r|)(|a_i| - m|_a|)}{\sqrt{\sum_{i=1}^n (|r_i| - m|_r|)^2} \sqrt{\sum_{i=1}^n (|a_i| - m|_a|)^2}} \quad (34)$$

where (| |) means absolute value and

$$m|_r| = \frac{1}{n} \sum_{i=1}^n |r_i| \quad (35)$$

$$m|_a| = \frac{1}{n} \sum_{i=1}^n |a_i| \quad (36)$$

⁶A.Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, 1965, chap. 7.

⁷D.G.Luenberger, Optimization by Vector Space Methods, John Wiley & Sons, 1969, chap. 2.

This cross-correlation can also be computed by the routine. This is accomplished by finding the magnitude of each r_i and a_i sample using CORDIC. Then the real part of r_i or a_i is replaced by the corresponding magnitude and the imaginary part is set to zero. The subsequent computations are identical for both real and complex cross-correlation. However, for real correlation the computed phase should be zero or near zero and only represents roundoff errors in the calculations. Hence, it is to be ignored.

For additional details refer to the flow diagram Figure 12. Most of the time required for correlation is spent in the loop which samples the sensors, increments frequencies and computes partial products for r . Since the origin frequencies of the two sensors are slightly different, it is necessary to retune the synthesizers between sampling r_i and a_i for the same grid point i . Once the loop is exited the remaining computation requires only about one second.

F. Major Shared Subroutines

1. Subroutine to Move Stepper Motors (A820_H-A8DD_H) - This subroutine moves each stepper motor by a specific amount which depends on the contents of three 2-byte memory locations labeled XTRAN, YTRAN, and ANGLE. Before the subroutine is called these locations are filled with 2's complement numbers. In the cases of XTRAN and YTRAN they are equal to the desired displacements of the x- and y-translation stages, respectively, in hundredths of a millimeter. In the case of ANGLE the number specifies the desired rotation in tenths of a degree.

The subroutine always operates the three motors one at a time in the order given above. Upon entry certain parameters are set up which are peculiar to the motor being operated. Then the program number is checked because Program D, whose function is to update the position display without moving the motors, has the same entry point. Next the desired number of steps is changed from 2's complement to sign-magnitude form. If the

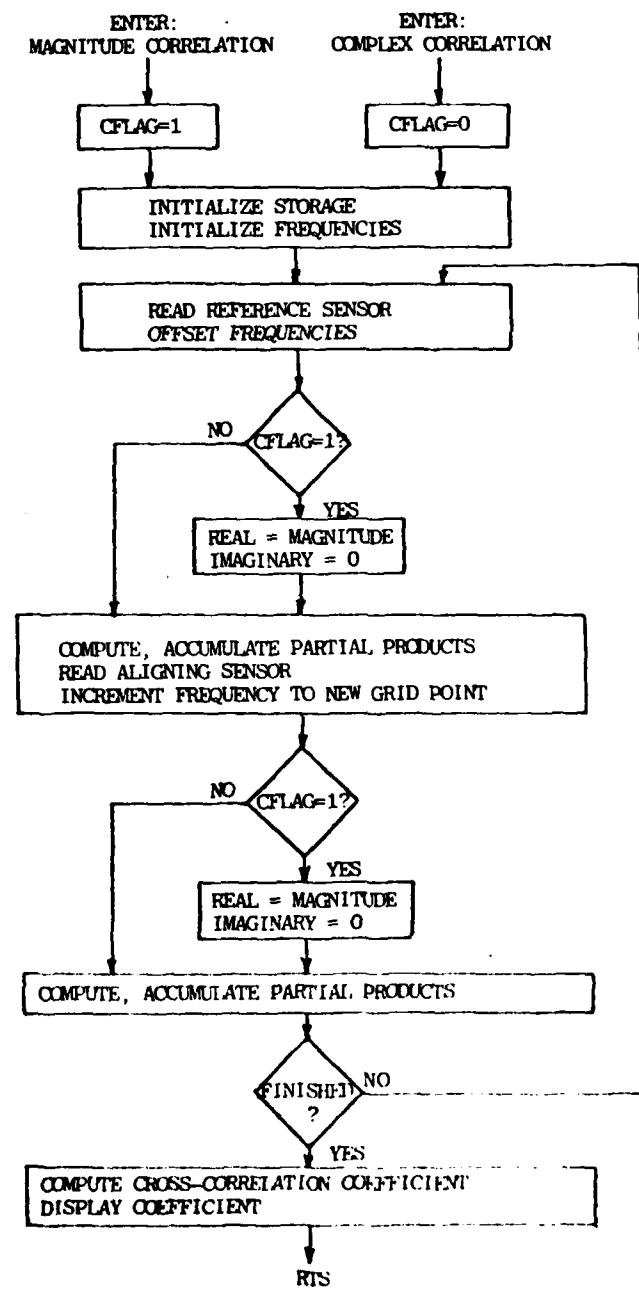


Figure 12 - Flow Diagram for Correlation Coefficient Subroutine

number is equal to zero the program goes directly on to the next motor.

Before the motor is operated a bit is either set or cleared in the PIA which establishes the direction of motion. The program operates by applying a given number (contained in Accumulator B) of clock pulses to the motor control and then decrementing the desired number of steps. Since the translation tables have a step size of .002 mm, the number of pulses is 5 for horizontal and vertical motion. For rotation it is 3 since the step size for the rotating stage is 1/30 degree. After each pulse which steps the motor the position display is updated, but only if the system is in a calibrated state.

When the desired number of increments has been counted down to zero, the subroutine returns to take care of the next motor. The last return is to the calling program.

2. Position Display Subroutine (AADE_H-AB93_H) - This part of the code is responsible for keeping track of the motion of the translation and rotation stages and updating the front-panel LED display accordingly. It is called by the stepper motor subroutine in the case of automated operation, or by the supervisor if the stepper motors are actuated manually. Entry is made with Accumulator A containing a mask which identifies the clock bit in the stepper motor control interface. The next bit to the left is the motor direction. In order to make up, right, and clockwise be the positive directions it was necessary to invert the direction bit for horizontal or rotational motion.

A total of five bytes is reserved in RAM for the position of each stage. Three bytes keep the step count in BCD form for the display, and two bytes keep it in 2's complement binary form. In the case of translation, the least significant BCD byte is incremented or decremented by 20 for each clock pulse, since the program step size is 5 times as great as the step size of the stage. For rotation the increment or decrement is

33. To prevent roundoff error from accumulating, 99 is rounded up to 100 and 1 is rounded down to zero.

The two most significant BCD bytes are changed from 9's complement to sign-magnitude form, converted to the 7-segment display code, and stored in the appropriate part of the display buffer. The display is refreshed by the interrupt service routine, located at AAA5_H.

This subroutine has another entry point at AB3D_H which is used by the image correlation routine to display the result of the computation. Entry here is made with a two-byte 9's complement BCD number in the A and B accumulators. This number is changed to sign-magnitude form and displayed.

3. Synchronous Sampler Subroutine (AE2A_H-AEBD_H) - This part of the program operates the DAS to sample the signals from the detector circuit boards in synchronism with the bias frequency and the 60 Hz line frequency. Single-bit inputs on the PIA's which control the synthesizers are configured to set flags internal to the PIA's when the 60 Hz line and the bias signal undergo positive- or negative-going zero-crossings. These functions are independent from the synthesizer control functions even though the same PIA's are used.

After clearing the memory locations where the signal values will be accumulated, the subroutine sets the interrupt mask so the time required for interrupt service will not disturb the synchronism of the sampling. It then waits for a transition of the 60 Hz line, which marks the beginning of a group of 16 consecutive signal samples taken at the positive and negative peaks of the signal. At the bias frequency of 1440 Hz, the positive and negative peaks are separated by 347 μ s, so the 16 consecutive samples take 5.5 ms, which is somewhat less than a half-cycle at 60 Hz. After the next 60 Hz transition another 16 samples are taken. This method distributes the samples evenly over a full cycle of the line frequency so that cyclic

variations of the image brightness do not affect the data. The interrupt mask is cleared between groups of samples so the interrupt service routine can refresh the front-panel LED display.

A software time delay between the bias reference transitions and the sample commands was adjusted experimentally to make the samples coincide with the signal peaks. The pair of signals which represent the output from each sensor are in phase, although their voltages represent two orthogonal vectors which describe the signal's phase as well as its amplitude. Two samples are taken from one of the signal pair at positive and negative peaks, followed by two samples from the other one of the pair, and so on. Alternate sampling of the two orthogonal signal components minimizes phase errors caused by short-term signal fluctuations. Sampling positive and negative peak values and accumulating their difference eliminates any dc offset associated with the signal and provides some additional narrowband filtering around the bias frequency.

Exit from the subroutine occurs with the two orthogonal signal vectors accumulated separately in four consecutive memory locations pointed to by the x register. The accumulation scales the 12-bit signal voltage samples up by a factor of 16 so that they each fill two bytes.

V. OPERATING INSTRUCTIONS

A. Operating Functions

The Image Alignment and Correlation System has been programmed to perform fourteen separate functions. The control programs for these functions are stored permanently in UV-erasable read-only memories in the microprocessor subsystem. The software is organized as a supervisor or main program and a master subroutine for each system function.

When power is first turned on, the supervisor enters a wait loop with dashes displayed on the front-panel LED readout. The dashes indicate that the system is ready to accept commands from the keypad directly below the readout.

Pressing a key at this point will result in the display of the corresponding program number, with two exceptions. They are "A", which is reserved for ABORT, and "E", which is used for EXECUTE. After entering a valid program number, pressing "E" will cause execution of that program to begin. Any time before "E" is pressed, entering a new program number will override the preceding entry. Pressing "E" initially will have no effect. Entering "A" at any time will cause a return to the supervisor, and the execution of any program in progress will be terminated.

Next is a tabulation of the fourteen program functions, followed by a description and instructions for each one.

TABLE 5
Pre-programmed System Functions

<u>Identifier</u>	<u>Description</u>
0	Graphic x-y plot (hard copy) of spatial frequency content of test image (i.e., the image to be aligned).
1	Graphic CRT display of spatial frequency content of test image.
2	Graphic x-y plot of spatial frequency content of reference (fixed) image.
3	Graphic CRT display of spatial frequency content of reference image.
4	Initial alignment and calibration using both sensors.
5	Calibration for re-alignment using test image only (single sensor).
6	Alignment after calibration using both sensors, including rotation.
7	Alignment after calibration using both sensors, without rotation.
8	Alignment after calibration using test image only, including rotation.
9	Alignment after calibration using test image only, without rotation.
A	Abort execution and return to supervisor.
B	Compute real image correlation coefficient using magnitude of spatial frequency data.
C	Compute complex correlation coefficient using complex spatial frequency data.
D	Display position of test image on LED readout (only after calibration).
E	Execute displayed program number.
F	Re-initialize parameters.

0 - X-Y Plot from Test Image - This program operates the accessory x-y plotter, making it draw on paper a pseudo three-dimensional graphic representation of the spatial frequency content of the test image. (The test image is the one on the left, viewed from the sensor modules.) Before executing this program for the first time, the instruction manual for the plotter (Hewlett-Packard model 8015B) should be read and understood. In addition, the following steps must be taken prior to execution:

- a. Connect the power cord on the x-y plotter to the 115V ac supply.
- b. Connect the interfacing cable from the x-y plotter to the system's electronics cabinet.
- c. Position a clean sheet of paper on the plotter. The use of paper furnished by the plotter manufacturer is recommended.
- d. Set up the plotter controls as follows:

Line - On

V/In - 1 for both x and y

Cal/Vernier - Cal for x; vernier for y

Chart - Hold after paper is in place

Servo - On (Note: Pen may move quickly.)

Pen - Lift

Reset/Sweep - Reset

X Inputs/X Time Base - X Inputs

- e. Place a pen of the desired color in the holder.
- f. Depress Zero Check for the y axis and adjust Zero so that the pen is directly over the lowest line on the chart grid. Repeat for the X Asix, placing the pen over the left-hand end of the grid. (Note: The pen may move quickly to the zero locations when the Zero Check button is pressed. Be sure its movement is not obstructed.)
- g. Last, remove the cap on the pen and lower the pen holder.

The program may now be executed. A complete plot takes about 14 minutes.

Replacing the pen cap after each plot will prevent drying of the felt tip and will prolong pen life. It is also a good idea to keep the plastic cover on the plotter when it is not in use. However, be sure the power to the plotter is off before replacing the cover.

Periodically the recorder's y-axis vernier gain adjustment should be checked, although it affects only the vertical size of the graph. The procedure is as follows:

1. Perform steps a. through f. above.
2. Turn on the graphics terminal.
3. Open the door of the electronics cabinet and press the right-hand one of the two small white buttons. An asterisk will appear on the CRT.
4. Using the terminal's keyboard, enter MEE80 and press the RETURN key. The terminal will respond by printing EE80 followed by a space and two hexadecimal digits.
5. Now enter FF and RETURN. The pen on the recorder should move quickly to the top of the paper.
6. Do not disturb the Zero knob, but adjust the Vernier knob to position the pen directly over the top line on the paper.
7. While holding the CTRL and SHIFT keys down together, enter K. The terminal will respond with an asterisk.
8. Either press the BREAK key on the terminal or the small white button on the left inside the door of the cabinet. The pen should move quickly to the bottom line of the paper and dashes should return to the front panel LED display. This completes the adjustment.

1 - Graphic CRT Display from Test Image - This program produces the same display as Program 0, but it appears on the accessory graphics terminal instead of the plotter. In this case, be sure that the interfacing cable from the terminal is connected

to the electronics cabinet, and that power is applied to the terminal. The terminal's ac power switch is located at the rear, on the right side as viewed from the front. Also check that both the transmit and receive baud rate switches on the rear of the terminal are set to 4800. Execution can then be started.

When program execution begins, the display, which is a storage CRT, will be erased. The program then labels the top of the display to show the x and y transducer frequencies at the starting point, which is the lower left-hand corner. The label also identifies the image being examined. The display covers a span of 7.6 MHz along each axis. A complete plot takes about 9 minutes. The terminal will sound an audible "beep" when the display is completed.

Except for the PAGE and BREAK keys, entries at the keyboard have no effect on this program. Pressing the PAGE key will erase the display. Execution will continue, but the terminal will print characters instead of drawing vectors. The only recourse is to abort execution and restart it.

Depressing the BREAK key causes a hardware reset which terminates any execution in progress and re-initializes the entire system. This applies to all of the fourteen available program functions as long as the terminal is connected and turned on.

The terminal should be turned on for a warm-up of several minutes before use. After prolonged periods of inactivity, the manufacturer recommends a 20-minute warm-up.

2 - X-Y Plot from Reference Image - All of the comments under Program 0 apply, except that the data is taken from the module which examines the reference image.

3 - Graphic CRT Display from Reference Image - All of the comments under Program 1 apply, except that the data is from the reference image.

4- Initial Alignment and Calibration, Both Sensors - In demonstrating alignment of the Test Image with the Reference Image, it is first necessary for the system to align the Test Image to a position which is defined as zero error. Program 4 performs this function. It begins by searching the spatial frequency domain of the reference image to select a set of up to 16 points which are associated with prominent components of the image's spatial frequency spectrum. The system retains the locations of these points in the spatial frequency domain, and uses them first to achieve a preliminary alignment in angle. It searches over a range of ± 6 degrees from the original orientation of the Test Image, and computes a cross-correlation coefficient based on the magnitudes of these sample points. After finding the angle where the correlation is greatest, the system uses the phase information from the same data points to align the Test Image laterally and vertically. It then alternately performs fine angular and translational adjustments, until the next computed translational correction falls below a preset threshold. At that point the system sets an internal flag which designates the calibrated state, and control returns to the supervisor.

Neither the CRT terminal or the x-y plotter are required for this program. The light boxes must be turned on with suitable patterns in place before execution is started. In addition, the two push-button switches on the electronics cabinet marked MAN/AUTO must be in the AUTO position. Best alignment performance is obtained after a period of at least 30 minutes to allow for thermal stabilization of the sensors. One of the plotter programs should be executed after the system is first turned on, to initiate the stabilization.

It is good practice to execute Program F immediately before starting this function, to insure proper initialization.

5 - Calibration for Re-alignment, Single Sensor - This program is similar to the preceding one, except that only the Test Image

is involved. The system may be calibrated with the Test Image in any position, and subsequent alignments will return it to that position. The selection of a set of data points in the spatial frequency domain proceeds as in Program 4, but in this case the magnitude and phase values at these points are simply stored in memory, and no rotation or translation of the image takes place.

The last two paragraphs under Program 4 above apply here also. However, the single-sensor mode of operation is more sensitive to thermal drift, so calibration should be done immediately prior to subsequent re-alignments using Programs 8 or 9.

6 - Alignment with Both Sensors, Including Rotation - Before
executing this program the system must be aligned for calibration using Program 4. Any attempt to run either Program 6 or 7 before calibration, or immediately after a system reset, will cause a return to the supervisor with the word "Error" shown on the front-panel LED display.

After the system has been aligned for calibration using Program 4, the Test Image can be displaced and rotated manually before this program is run to demonstrate re-alignment. Manual control of the stepper motors is effected with the two front-panel buttons labeled MAN/AUTO in the "out" (MAN) position. The step, direction, and slew buttons, and the x/y button in the case of rotation, can then be used to move the Test Image to the desired starting point. The position on each axis will be shown on the LED display as each stepper motor is activated. The angular misalignment should not be made more than 6 degrees, so as not to exceed the search range of the alignment program. The allowable translational misalignment depends on the image, but displacements up to ± 5 mm have worked well in our tests.

As with any of the alignment programs (4, 6, 7, 8, or 9), the MAN/AUTO switches must be returned to AUTO before this

program is executed. Failure to do so will result in a program halt with "Error" displayed. The program may be restarted by pressing the "E" key after resetting the stepper motor controls to AUTO.

While the program is running, the LED display will show the position of the Test Image. When no further correction is indicated, execution will stop and dashes will return to the display. The final errors may be examined by pressing "D" followed by "E" on the front-panel keypad. (See section on "D.")

If the accessory CRT terminal is connected and turned on in advance, it will show a graphic display of the translation of the Test Image as alignment progresses. Initially the display will have the x and y axes with a very small square at their intersection. This square represents translational error bounds of 0.1 mm (100 μ). The display encompasses an area in xy-space of about 24 mm by 18 mm. When the translational errors have been reduced to less than 1 mm in x and 0.5 mm in y, the display will be erased and replaced by a new one which is magnified 16x. The display does not give any information as to the angular position of the image.

7- Alignment with Both Sensors, Without Rotation - This program is similar to the preceding one except that the Test Image is re-aligned only in translation. Omitting the angular correction decreases the running time, and this mode also can be used to study the effects of constant angular errors on the translational corrections.

Except for those which refer specifically to rotation, all of the comments under Program 6 apply here also. The most accurate translational re-alignment will take place when no angular offset is introduced after calibration.

8 - Alignment with Single Sensor, Including Rotation - This function is again similar to Program 6, except that only the Test Image is used. Program 5 must be run before this program is executed. Otherwise, the "Error" message will be displayed

and control will return to the supervisor. The most accurate results will be obtained when re-alignment immediately follows calibration.

Except for the difference in program numbers, the discussion under Program 6 applies here also.

9 - Alignment with Single Sensor, Without Rotation - This program is the only remaining permutation of the alignment functions. It must be preceded by Program 5 for calibration. Except for the lack of rotation, it is the same as Program 8.

A - Abort - Pressing this key at any time will stop the execution of any program which is in progress. The front-panel LED display will fill with dashes, showing that control has returned to the supervisor. In general, it is good practice to run Program F following an abort.

B - Correlation of Magnitudes - This program computes a normalized cross-correlation coefficient from the magnitudes of a large set of samples in the spatial frequency domain of both sensors. The samples are taken on a square grid at intervals of 100 kHz, over an area 6 MHz square in the transducer frequency domain. The corresponding area in the spatial frequency domain is 20 cycles square, and is somewhat smaller than the area covered by the plotter programs.

Mathematically, the correlation coefficient can be expressed as Equation 34 (page 63).

In order for the correlation program to give the correct result, the most recent calibration program executed must have been Program 4 (i.e., the one involving both sensors). Normally this would be done anyway, to insure that the degree of correlation is being measured between two images which are properly aligned. If the effect of misalignment is to be studied, the Test Image can be displaced manually after the initial alignment has been completed.

This program takes about 14 minutes to execute. At its completion the correlation coefficient will appear on the LED

display, and on the CRT terminal if it is connected. The terminal will also sound a "beep" to alert the user. The first number displayed is the result. The second number, labeled "Phase" on the CRT, should be very small and has no meaning in this case since a real number is computed.

C - Complex Correlation - In this case a complex normalized cross-correlation coefficient is calculated, using the same data grid as in Program B. Here the expression is the same as Equation 31 (page 63). The result is in polar form, with the phase given in degrees.

Running time is about 14 minutes, and a "beep" will sound from the terminal upon completion.

Both of the correlation computations reflect the response of the sensors as well as the content of the images. The complex computation takes the phase of the samples into account, and the sensors are not matched as well in phase as they are in magnitude. Therefore, the complex correlation value tends to be smaller than the value computed from the magnitudes alone, for a given pair of images.

D - Position Display - This function uses the LED readout to display the position of the Test Image relative to its calibrated position. The first two numbers are the lateral and vertical translations in millimeters, respectively, and the third number is the angular position in degrees.

The displayed positions will always be zero upon initial turn-on of the system. In addition, Programs 4, 5, or F will clear the position counters, returning the displayed values to zero. When the system is in an uncalibrated state (i.e., neither 4 nor 5 has been run after initialization) manual operation of the stepper motors will not affect the position counters or the display. In the calibrated state, the position counters and the display will track any manual or programmed movement of the Test Image.

E - Execute - Any time a program number shows at the left-hand

end of the LED display, pressing the "E" key will start execution of that program. Pressing "E" when the dashes are displayed has no effect.

F - Re-Initialize Parameters - When the system is first powered up, a hardware reset vectors the processor to a sequence of instructions which, among other things, sets a number of program parameters in read-write memory. An example is the counters which keep track of the translation and rotation stages. These counters are set to zero initially, but can be changed during operation of the system.

Program F returns all of these parameters to their initial values. It was included to allow for the possibility that parameter values could be changed selectively by using the processor's internal monitor through the CRT display keyboard. Such changes are not recommended unless the user understands both the software which implements the alignment algorithm and the processor's internal FANTOM-II monitor. The monitor is documented in the Commercial Data, and the software listings appear in Appendix B.

Although the system has received many hours of testing, the possibility still exists that the software has some "bugs" that have not been identified. For this reason it is a good practice to use Program F immediately prior to the execution of any other program except those which follow a calibration. In other words, do not precede Programs 6 through 9 by Program F.

VI. REVIEW OF THE SYSTEM'S DEVELOPMENT

Our technical proposal, on which this contract award was based, reveals that the final form of the system as it was delivered to ETL is remarkable similar to that originally envisioned, in spite of a number of significant technical problems which were encountered subsequently. The block diagram in the proposal is nearly identical to the current one in Section II. It was clear at the outset that the system should have microprocessor control. An image alignment algorithm which used the magnitude of the Fourier transform to achieve angular alignment was also seen as very probable, since the magnitude of the transform theoretically is insensitive to image translation. We knew that the phase of transform components would be the key to translational alignment, but the relationships involved turned out to be less clear than anticipated.

The first work undertaken on this program was the selection of a suitable microprocessor subsystem. Originally it had been planned to concentrate first on the DEFT sensors and their surrounding circuitry. However, at that time a design for a new, higher-resolution DEFT sensor operating near 100 MHz was about to be tried, and we wanted to use it in this system if it could be proven in time. Therefore, the sensor work was exchanged in the schedule for work with the microprocessor. This effort involved the circuit design of interfaces between the processor and the stepper motors, the frequency synthesizers, the data acquisition system, the x-y plotter, the CRT display, and the front-panel LED display and keypad along with programming to support each of these interfaces. All of this was accomplished with relative ease.

Over a period of time, one of the frequency synthesizers malfunctioned intermittently, and the cause was traced to defective plated-through holes on its circuit board. It was

finally replaced by the manufacturer. Unfortunately, we know of no other source for a similar product. However, the problem has not recurred.

Two major factors which were not well established until later in the program were the format of the images and their means of illumination. The original discussions on this matter ranged from back-lighted 35 mm transparencies to opaque photographic prints.

When it was decided to use front-lighted opaque images in the system, the design of the alignment fixture could be finalized, and it was fabricated. At about the same time it became clear that the 100 MHz sensor could not be perfected in time for inclusion in the system, so sensor modules were built with the 35 MHz sensor with which we had accumulated a fair amount of experience. In the meantime, the first version of the alignment algorithm had been programmed, so it became possible to try the image alignment function of the system. These first tests were encouraging, but they did not show the degree of alignment accuracy we were seeking.

As program debugging and system testing proceeded we became aware of a number of previously unknown factors which bore on the performance of the alignment algorithm. For example, the phase of the sensor output is approximately a linear function of the difference between the two SAW frequencies, with a proportionality constant of about 1 degree per kHz. This phase function is in addition to the phase which the image imparts to the transform. A phase change of 1 degree at a point 3 MHz away from the origin in the frequency domain corresponds to an image displacement of only 3.5μ at the sensor, or 70μ at the image with 20 \times demagnification. Therefore, much greater significance became attached to the relatively small differences in the SAW frequencies which identify the spatial frequency origins of the two sensors. It became necessary to measure these frequencies accurately so the alignment program could use

them to measure the phase of each sensor at its transform origin. These phases were then applied as corrections to the phases measured at other points in the spatial frequency domain, since theoretically the phase of the transform at the origin should be zero for any real image.

Even after these refinements were incorporated into the software, the alignment accuracy was not satisfying. Errors in translation were typically 0.2 mm and rotational errors were typically 0.3 degree. Although the delivery date specified in the contract was very close, we felt that better performance could and should be obtained. Therefore we decided to ask for a 90-day extension to the contract at no additional cost to the government.

In this request for an extension we proposed changes in the wording of the work statement so that the required alignment accuracy would be quantified. In addition, we suggested that image correlation be computed from the respective transforms rather than presented subjectively by means of the plotter or graphics display. The contract extension and the changes in wording were granted subsequently.

In the intervening time several changes were made in order to increase the signal-to-noise ratio from the sensors and to reduce or eliminate any suspected source of error. One factor which had been disturbing was the lack of contrast obtained with front-lighted opaque images. Several methods of making image patterns were tried, and even the most seemingly non-reflective surface tended to scatter back enough light to result in signals that were less than satisfactory. Therefore we decided to modify the alignment fixture to include light boxes so that transparencies could be used.

In order to get the highest brightness consistent with reasonable power and heat dissipation levels, fluorescent lamps were used in the light boxes. Unfortunately, the light

output of a fluroescent lamp varies considerably over the power line cycle, and this variation modulates the sensor output. This modulation would have introduced a significant source of error in the data. In order to avoid this error it became necessary to phase-lock the sensor bias voltage to the power line frequency and to synchronize the sampling of the signals with the line frequency also. This procedure averages a number of samples over a complete cycle of the power line frequency so that each data point is sampled with the same apparent light level.

As additional steps to increase the available signal-to-noise ratio, the f/1.4 lenses originally used on the sensor modules were replaced by f/0.85 lenses, and the sensor bias voltage was increased by a factor of 5. Typical signal-to-noise ratios at that point were of the order of 55 dB for prominent spatial frequencies.

Further improvements included a re-design of the layout of the rf distribution circuit board for better isolation and shielding, and a change in the synchronous sampling subroutine to interleave the "real" and "imaginary" samples. Both of these steps improved the accuracy of the data, particularly in regard to its phase.

With these improvements the alignment accuracy of the system exceeded the specification in the contract as modified, and it was delivered to ETL on schedule.

In order to compensate in some way for the extra time required to complete the system, a feature was added to the graphics display which portrays the motion of the test image as alignment progresses.

There is little to present in the way of study results on this contract since most of the work has dealt with hardware design and its practical problems. The exception to this is the development of the alignment algorithm, which is presented in Section III.

VII. CONCLUSIONS

The Image Alignment and Correlation System has demonstrated the application of DEFT technology to the problem of aligning two identical images in translation and rotation. The system uses a highly developed image-adaptive algorithm which exploits the spatial frequency analysis capability of the DEFT sensor. The alignment accuracy of the system is image dependent, but with high contrast images having prominent spatial frequency features, the accuracy approaches the resolution of the translation and rotation stages. The accuracy is limited by the signal-to-noise ratio of the image's spatial frequency components and by mismatch between the two sensors. However, the alignment algorithm makes corrections for these differences wherever possible.

In the course of developing this system, new information has been obtained regarding the use of an ac bias with the DEFT sensor, and on circuitry for processing the sensor's output signal coherently. This information has advanced the state of the art in DEFT applications, and will be of value to any related future development.

The alignment algorithm, its implementation in software, and particularly its refinement to suit the characteristics of the sensor, also represent a significant achievement in the application of DEFT technology. Especially noteworthy are the image-adaptive properties of that part of the program which selects the most useful spatial frequency data from the image.

The use of the method of least squares to provide translation offset estimates is sufficiently general so that alignment of rather arbitrary images should be possible when more sensitive DEFT sensors become available.

This system has shown that the spatial frequency information provided by the DEFT sensor can be used to actually perform a function, as well as being made available for interpretation and analysis.

VIII. RECOMMENDATIONS

The Image Alignment and Correlation System is most in need of improvement in the areas of operating speed and image dependence. Speed of execution was not a primary consideration in the design of the system, and it would have been impossible to assess this factor accurately prior to the development of the alignment algorithm. However, now that the system is complete and some experience has been gained with it, a reduction in execution times clearly would be a desirable improvement.

Speed of execution is limited primarily by the phase-locked loop frequency synthesizers in the system, which require about 30 ms for settling after a frequency change is commanded. Modifying the synthesizers to reduce their settling time by a factor of 10 would make a significant improvement in the operating speed of the system. There are probably areas of software which could be improved in regard to execution times. However, in the absense of a dc-powered light source, the need for data sampling which is synchronous with the ac line frequency puts a lower limit of about 16 ms on the time required for each signal sample. This factor alone accounts for 13% of the execution time for the image correlation function, for example.

The dependence of the system on particular types of images is based on the sensitivity of the DEFT sensors. Higher sensor output would allow operation with images having lower contrast or less prominent spatial frequency features. When improved sensors become available, they could be considered for retrofit into the system, and modifications to reduce execution times could be made concurrently.

AD-A091 840 DEFT LABS INC EAST SYRACUSE NY
IMAGE ALIGNMENT AND CORRELATION SYSTEM. (U)
JUL 80 S E CRAIG, A L MOYER

F/6 9/2

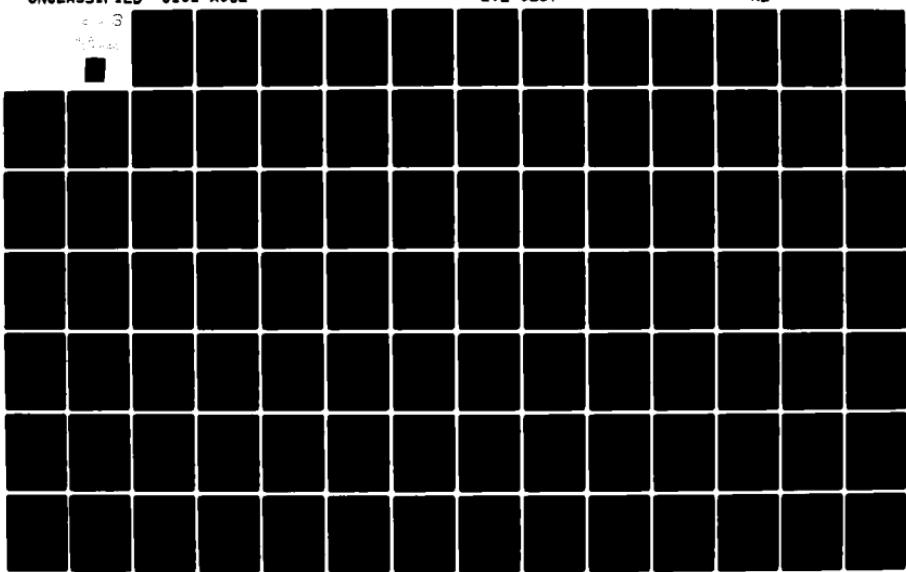
UNCLASSIFIED 0101-A002

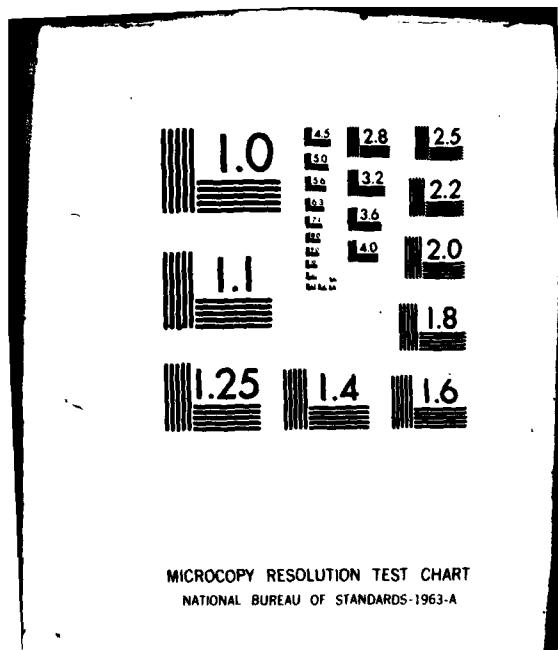
ETL-0237

DAAK70-78-C-0217

NL

c - 3
4 A, 100





IX. REFERENCES

1. J. W. Goodman, Introduction to Fourier Optics, McGraw-Hill 1968, chap. 2.
2. D. L. Kreider, et al., An Introduction to Linear Analysis, Addison-Wesley, 1966, chap. 7.
3. D. J. Luenberger, Introduction to Linear and Nonlinear Programming, Addison-Wesley, 1973, chap. 7.
4. J. E. Volder, "The CORDIC Trigonometric Computing Technique," IRE Trans. Comp. Sept. 1959, pp. 330-334.
5. D. J. Wilde, Optimum Seeking Methods, Prentice-Hall Inc. 1964.
6. A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, 1965, chap. 7.
7. D. G. Luenberger, Optimization by Vector Space Methods, John Wiley and Sons Inc., 1969, chap. 2.

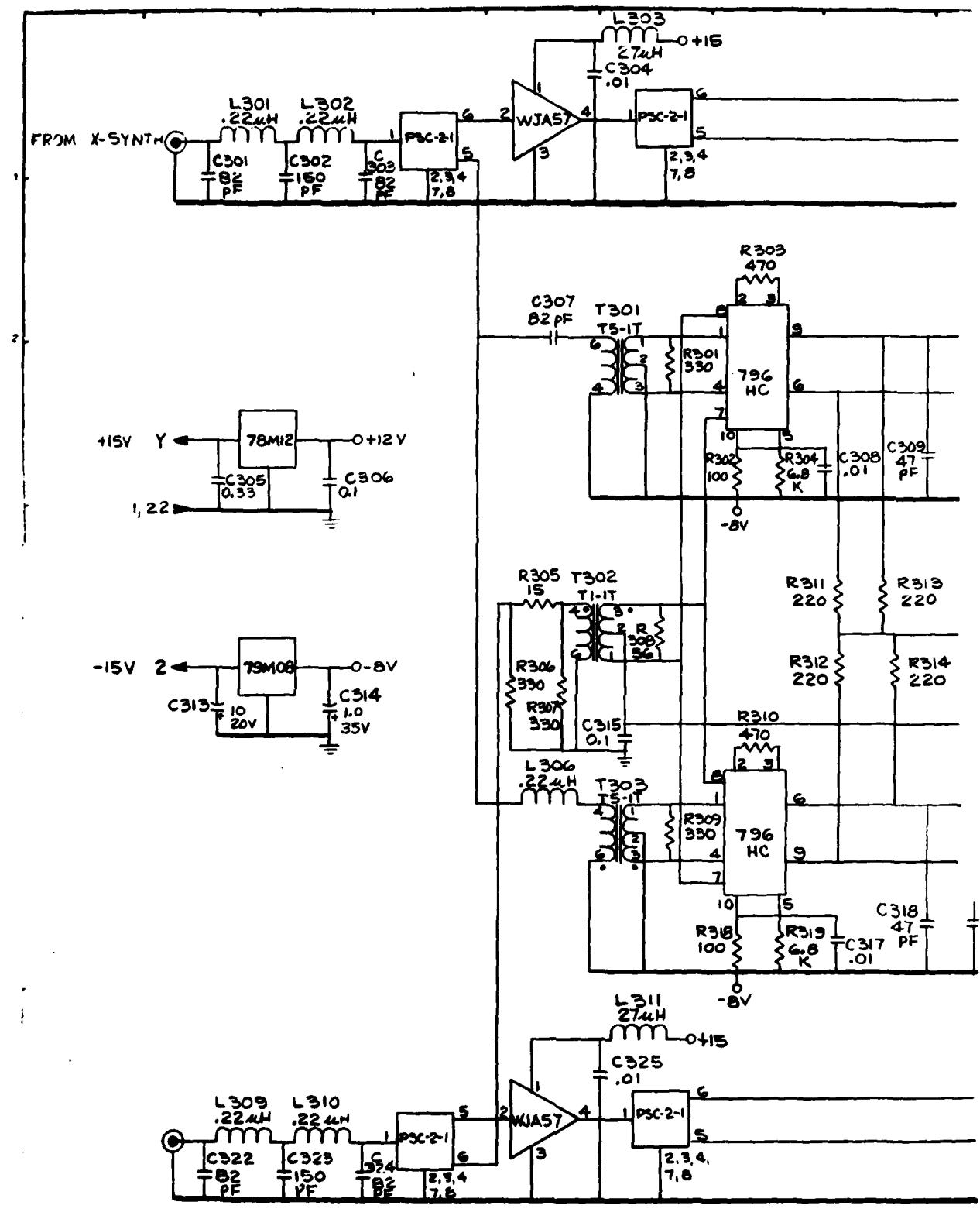
X. BIBLIOGRAPHY

1. Kowal, S.T. et al, DEFT: Advanced Structures and Applications, Report No. TR-79-12, Dept. of Electrical and Computer Engineering, Syracuse University, Syracuse, N.Y., Prepared for US Army Night Vision Laboratory, Fort Belvoir, VA 22060, Grant No. DAAG53-76-C-0162, 30 June 1979.
2. Kowal, S.T. et al, Two-Dimensional Direct Electronic Fourier Transform (DEFT) Devices - Analysis, Fabrication and Evaluation, Report No. TR-77-5, Dept. of Electrical and Computer Engineering, Syracuse University, Syracuse, N.Y., June 1977.
3. Kornreich, P.G. et al, "DEFT: Direct Electronic Fourier Transforms of Optical Images", Proc. IEEE, Vol. 62, August 1974, pp. 1072-1087.

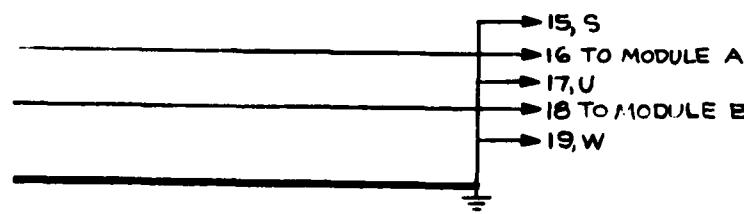
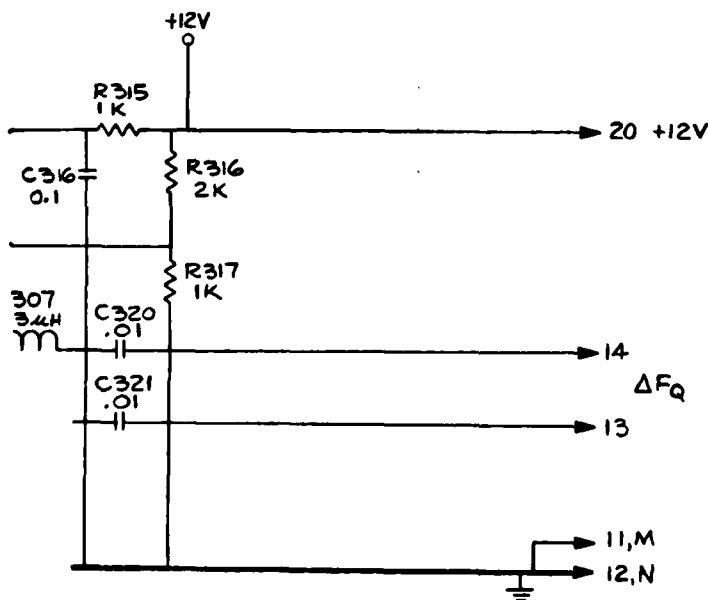
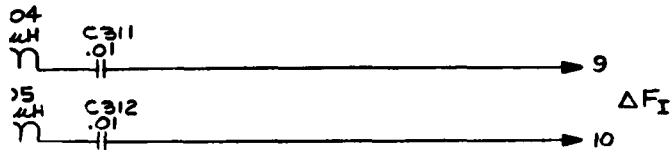
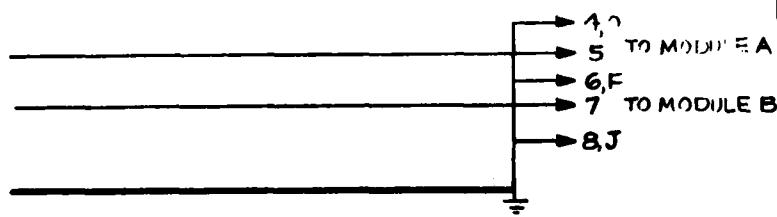
APPENDIX A

This section shows schematic diagrams for the various units described in Section II.

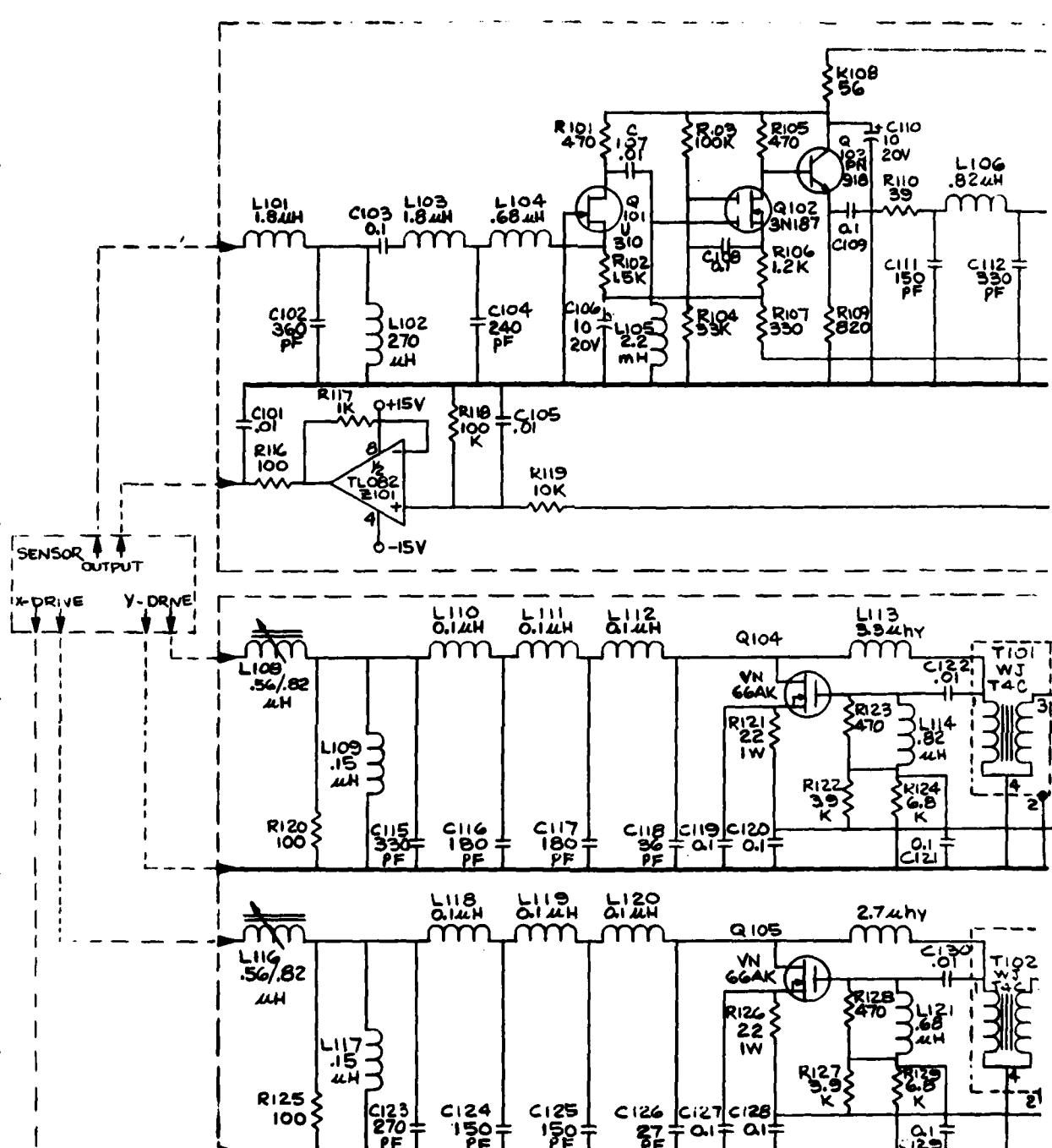
<u>Unit</u>	<u>Page</u>
RF Distribution PC Board	A-2
DEFT Sensor Module	A-4
Bias Generator PC Board	A-6
Synchronous Detector PC Board	A-8
RTI-1220 Buffer Board	A-9
System Wiring Diagram	A-10

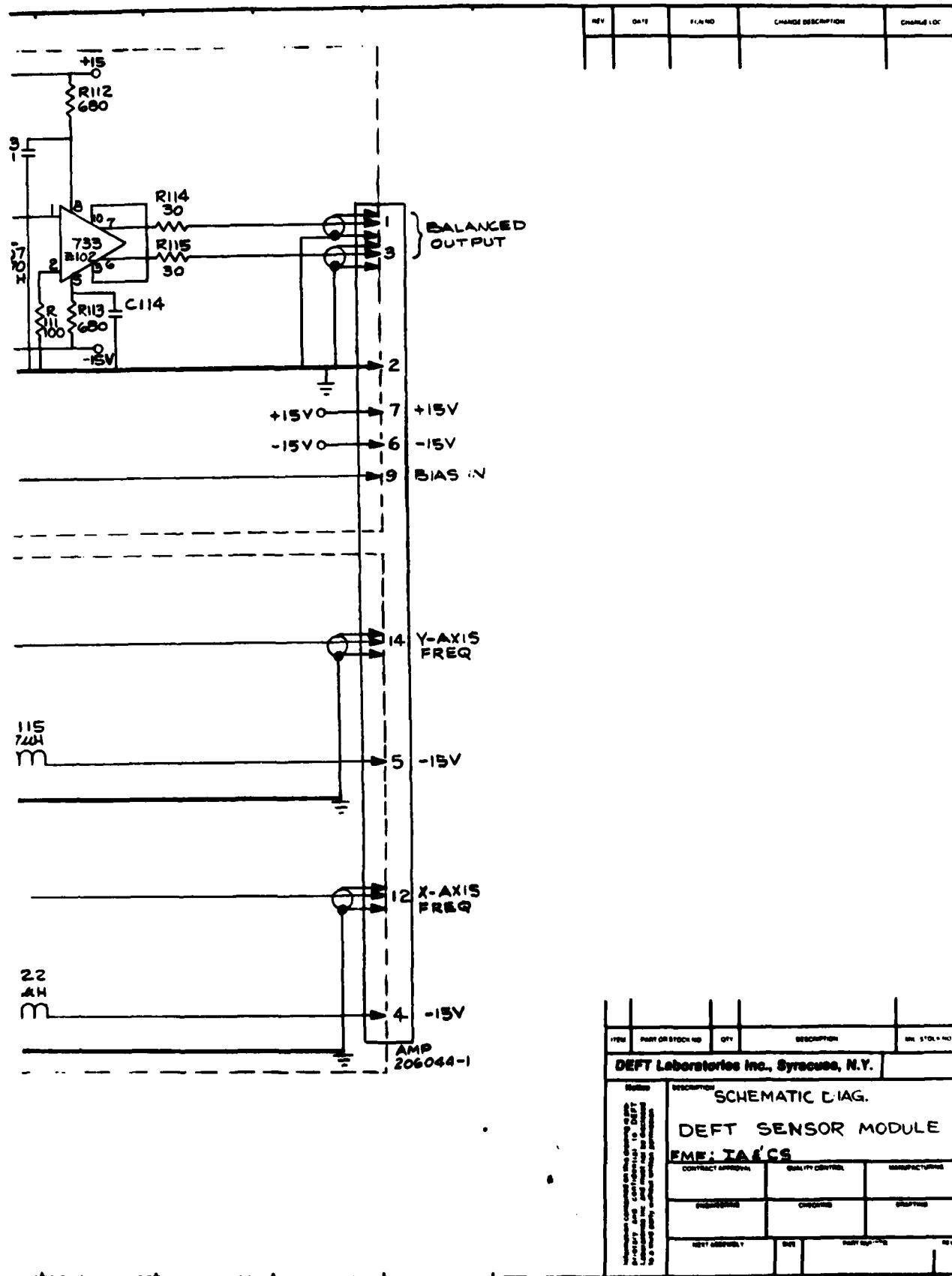


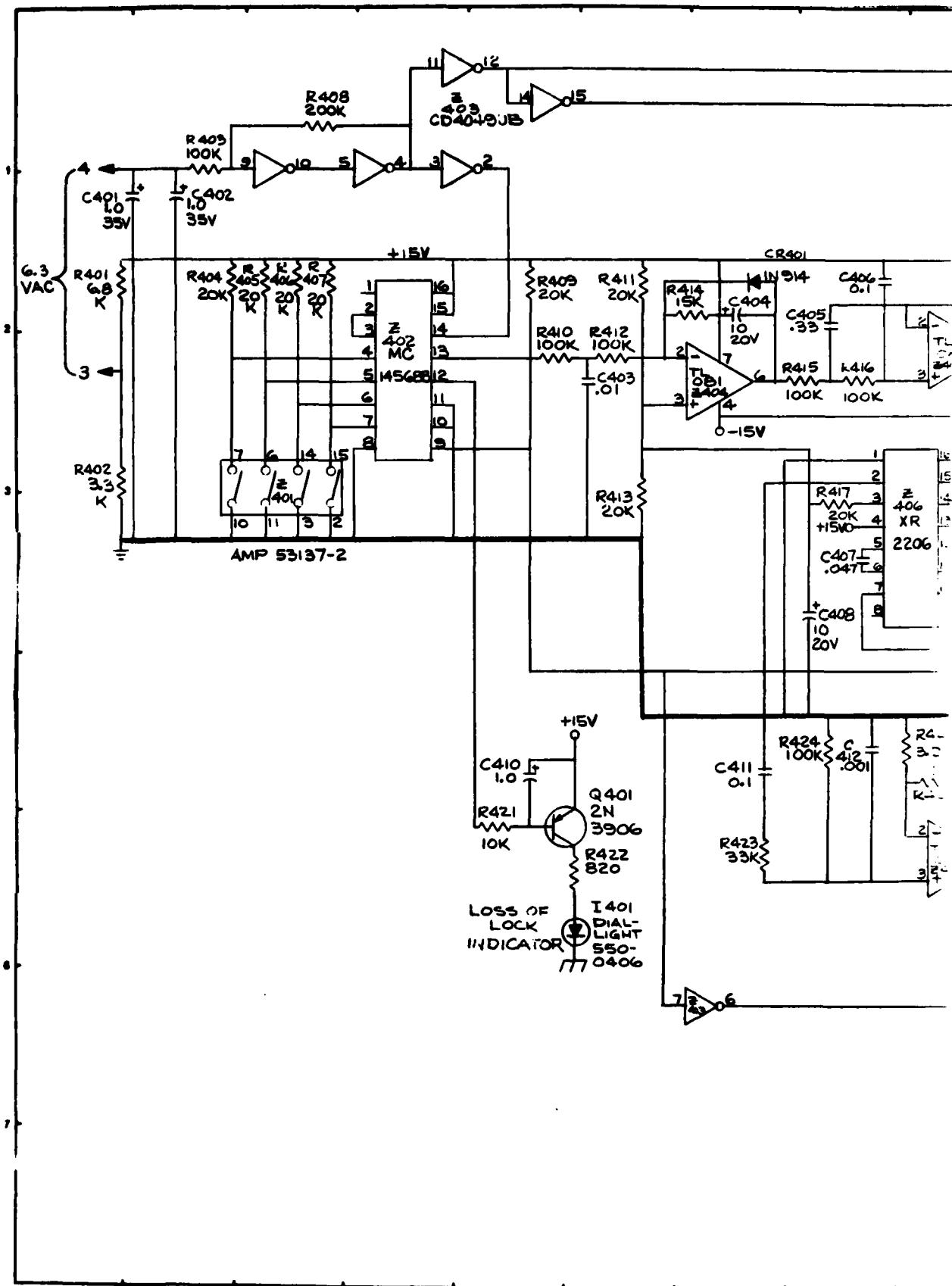
ITEM	PART OR STOCK NO.	QTY	DESCRIPTION	REMARKS

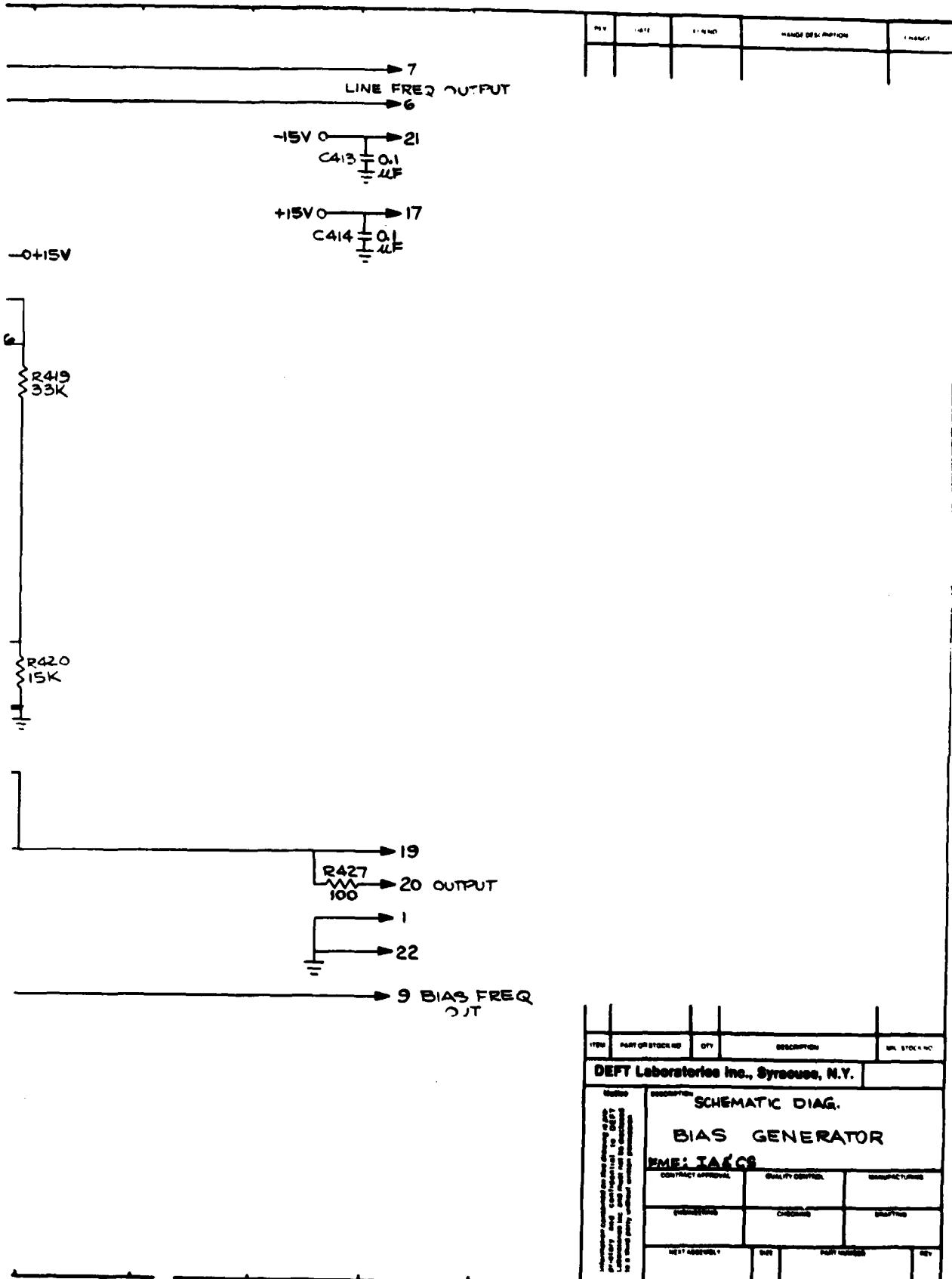


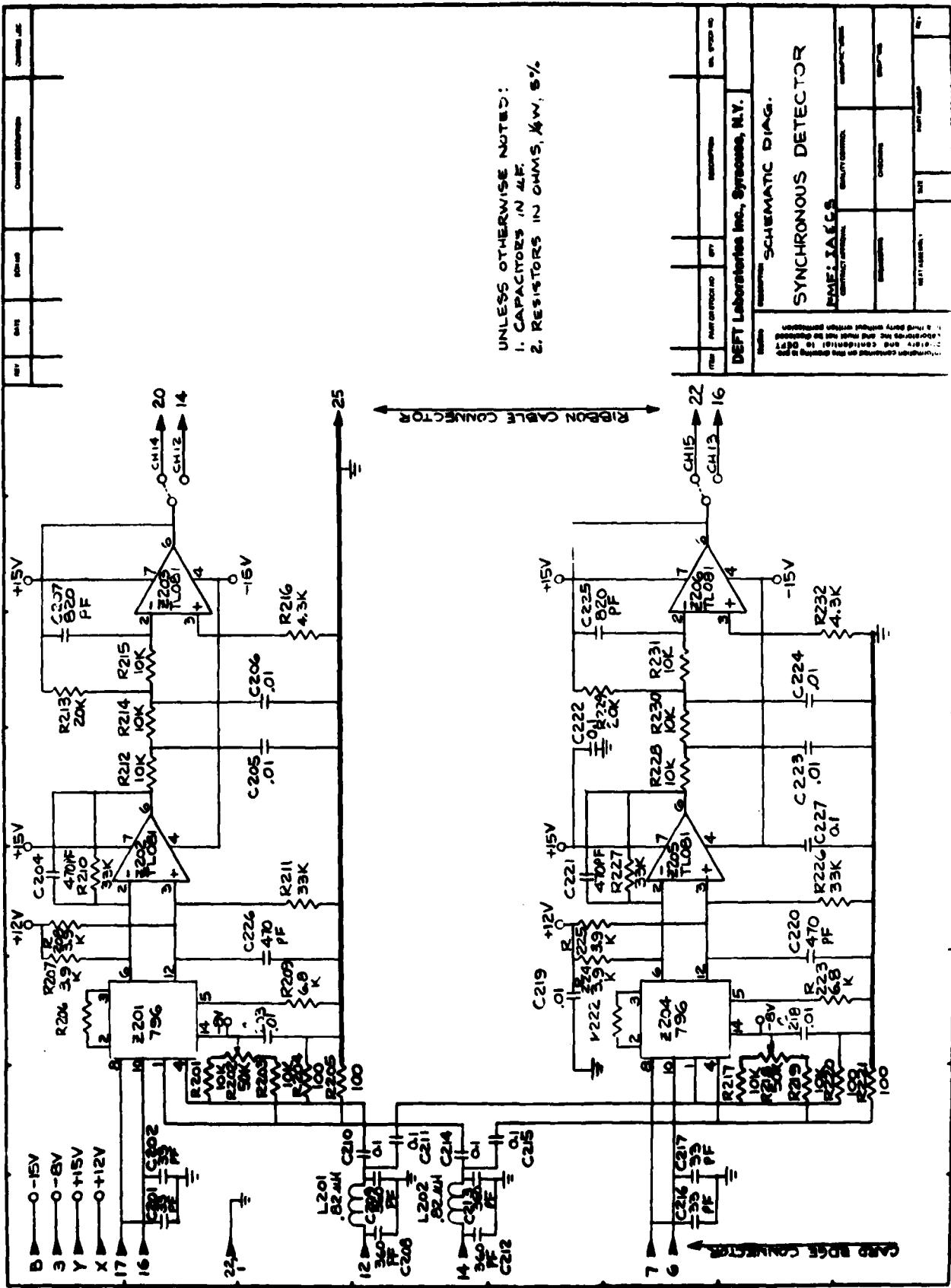
ITEM	PART OR STOCK NO.	QTY	DESCRIPTION	REMARKS
DEFT Laboratories Inc., Syracuse, N.Y.				
REVISION DATE REVISION DATE REVISION DATE	DESCRIPTION: LOGIC DIAG.			
	RF DISTRIBUTION			
	FME: TAE'S			
CONTRACT APPROVAL		QUALITY CONTROL		MANUFACTURE
PRODUCTION		CHECKED		DRAFTING
TEST APPROVAL		DATE		PART NUMBER

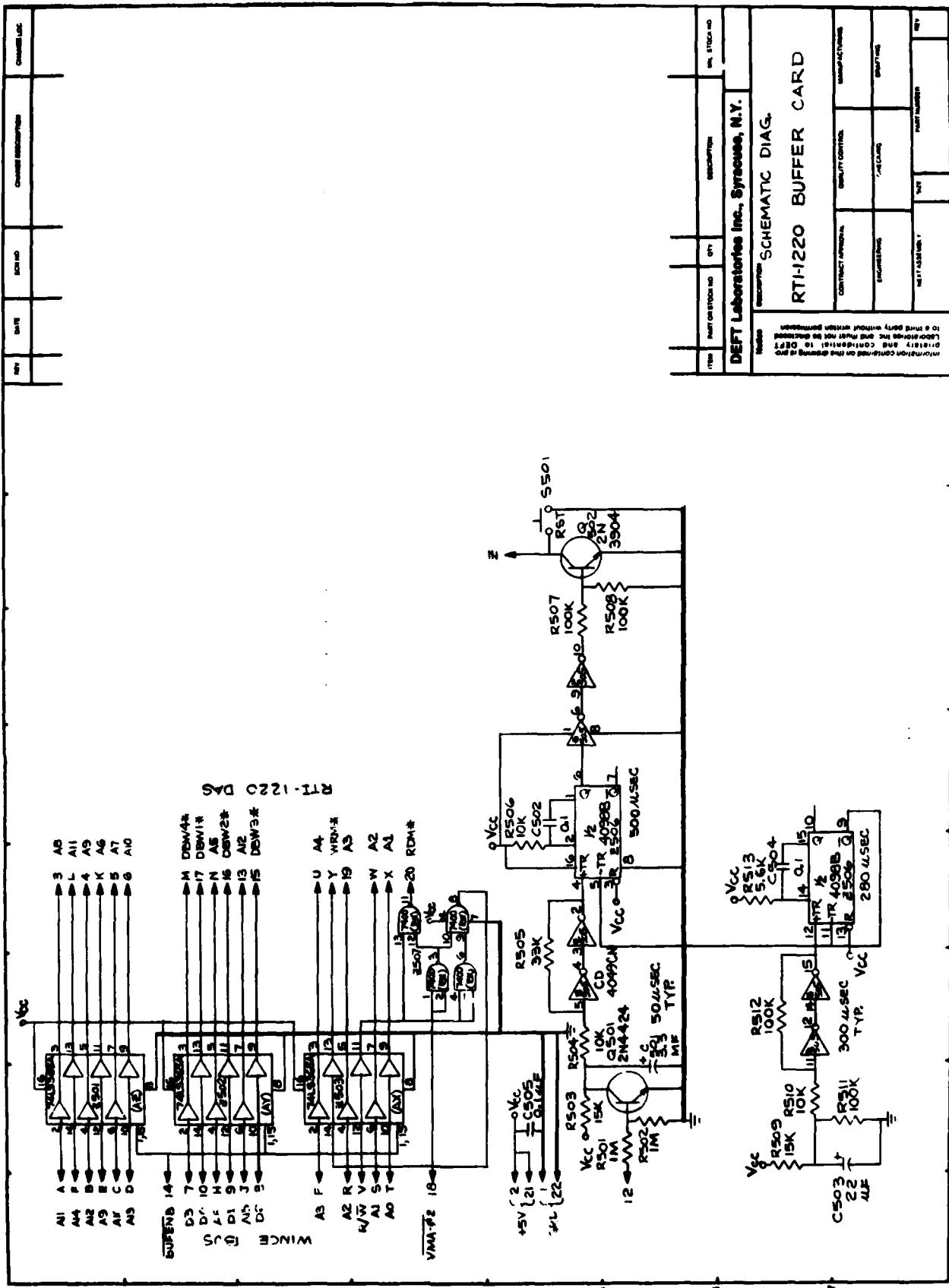


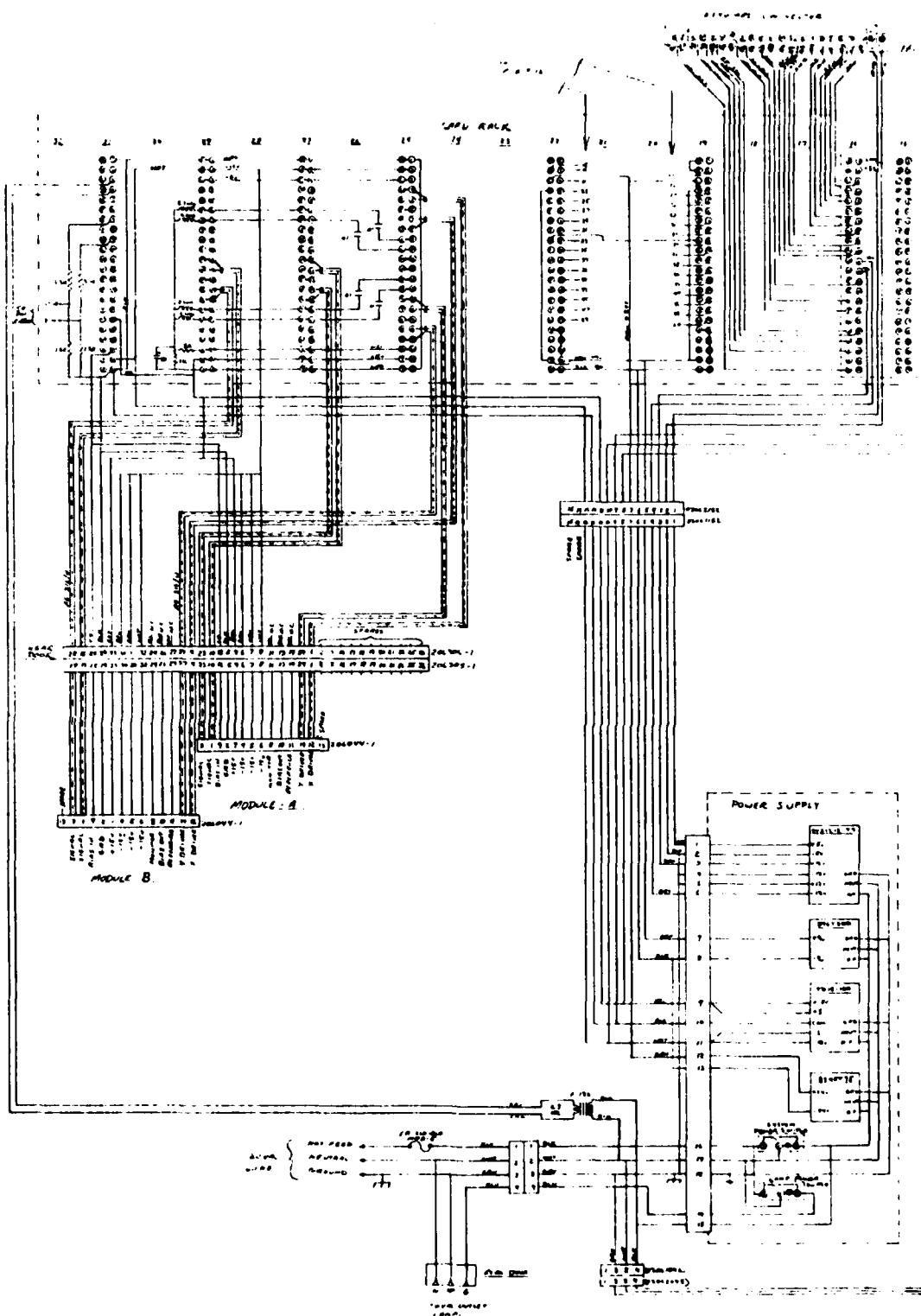


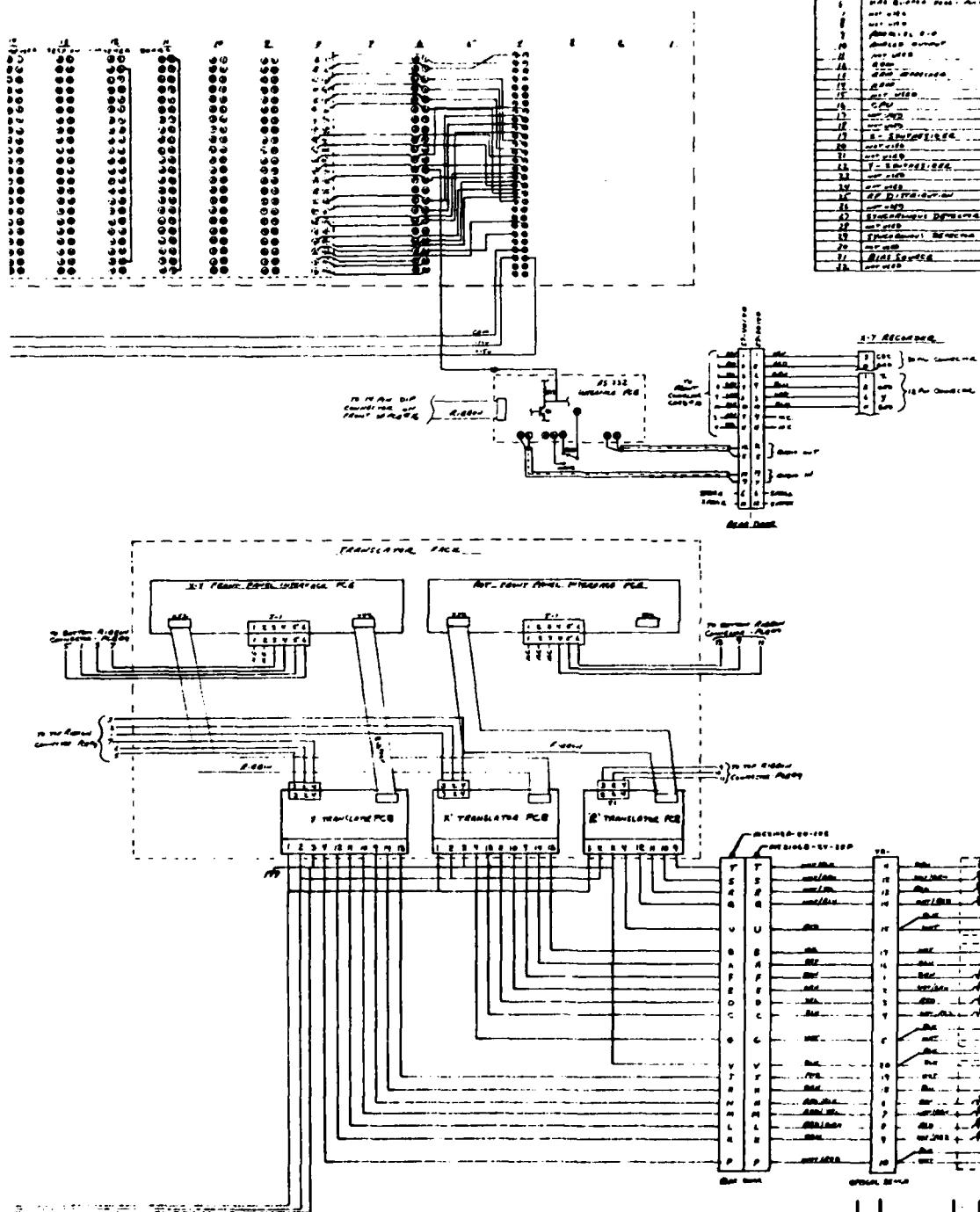












AP-82
1) NO DEMON NO CONCERTO
2) NO CONCERTO = NO DEMON DEMONIC ANGEL

A-11

APPENDIX B

This section contains listings of all of the software furnished with the Image Alignment and Correlation System except for that purchased from Wintek Corp. All addresses and opcodes are hexadecimal. In the operand column of the statements, the following symbols are used:

\$	Hexadecimal Prefix
%	Binary Prefix
H	Hexadecimal Postfix
D	Decimal Postfix
B	Binary Postfix
#	Denotes Immediate Addressing Mode

CORRELATION COEFFICIENT

			SUBROUTINE CORRELATION COEFFICIENT
			SCANS TRANSFORMS AND COMPUTES THEIR
			COMPLEX CORRELATION COEFFICIENT
1800		ORG	\$1800
1800	TEMP1	RMB	1
1801	TEMP2	RMB	1
1802	TEMP3	RMB	1
1803	TEMP4	RMB	1
1804	TEMP5	RMB	1
1805	TEMP6	RMB	1
1806	BCD1	RMB	1
1807	BCD2	RMB	1
1808	CTR	RMB	1
1809	CTR1	RMB	1
1824		ORG	\$1824
1824	COR1	RMB	1
1825	COR2	RMB	1
1826	COR3	RMB	1
1827	COR4	RMB	3
182A	COR9	RMB	1
182B	COR10	RMB	1
182C	I	RMB	1
182D	J	RMB	1
182E	ID	RMB	1
182F	JD	RMB	1
1830	UI1	RMB	1
1831	UI2	RMB	1
1832	VJ1	RMB	1
1833	VJ2	RMB	1
1834	MODE	RMB	1
1835	DTIME	RMB	1
1836	JSTART	RMB	1
1837	BDRY	RMB	1
1838	BDRYD	RMB	1
1839	NBDRY	RMB	1
183A	BDRYPT	RMB	1
183B	FIRST	RMB	1
183C	SENSOR	RMB	1
183D	NSAMP	RMB	1
183E	LOGS	RMB	1
187A		ORG	\$187A
187A	UP	RMB	4
187E	UP	RMB	4
1882	UV	RMB	4
1886	VV	RMB	4
188A	UU	RMB	4
1856	CFLAG	EQU	\$1856
1857	CCFLAG	EQU	\$1857
1E40	ARRAY	EQU	\$1E40
BB68	PULL4	EQU	\$BB68
BB8E	PUSH44	EQU	\$BB8E
BBB4	PUSH41	EQU	\$BBB4
BBB1	PUSH42	EQU	\$BBB1
B9A0	DELAY1	EQU	\$B9A0
BB17	BI\$BCD	EQU	\$BB17
BBF4	TUNE	EQU	\$BBF4
BCOD	RDDEFT	EQU	\$BCOD

CORRELATION COEFFICIENT

8000		MATH	EQU	\$8000	
B0D6		CORDIC	EQU	\$B0D6	
E000		LEDBFR	EQU	\$E000	
ED2F		DISPTR	EQU	\$ED2F	
AB66		FIXSGN	EQU	\$AB66	
AEF7		CORCRT	EQU	\$AEF7	
	*				
A000			ORG	\$A000	
A000	86 01	CORMAG	LDAA	\$1	MAGNITUDE CORRELATION
A002	B7 18 56		STAA	CFLAG	
A005	20 03		BRA	CR1	
A007	7F 18 56	CORCOM	CLR	CFLAG	COMPLEX CORRELATION
A00A	CE 00 00	CR1	LDX	\$0	
A00D	BF 18 00		STS	TEMP1	
A010	8E 1E 6F		LDS	\$ARRAY+47D	
A013	4F		CLRA		
A014	36	CR2	PSHA		
A015	08		INX		
A016	8C 00 30		CPX	\$48D	
A019	26 F9		BNE	CR2	
A01B	BE 18 00		LDS	TEMP1	
A01E	7F 18 57		CLR	CCFLAG	
A021	CE 39 26		LDX	\$\$3926	
A024	FF 18 30		STX	UI1	
A027	CE 30 52		LDX	\$\$3052	
A02A	FF 18 32		STX	VJ1	
A02D	BD BB F4		JSR	TUNE	
A030	CE 01 00		LDX	\$\$0100	
A033	FF 18 3D		STX	NSAMP	
A036	86 0A		LDAA	\$\$A	
A038	B7 18 35		STAA	DTIME	
A03B	BD B9 A0		JSR	DELAY1	
A03E	86 FF	CR3	LDAA	\$\$FF	
A040	B7 18 3C		STAA	SENSOR	
A043	BD BC 0D		JSR	RDDEFT	REF
A046	FE 18 24		LDX	COR1	
A049	FF 18 8A		STX	UU	
A04C	FE 18 26		LDX	COR3	
A04F	FF 18 8C		STX	UU+2	
A052	BD BC 0D		JSR	RDDEFT	REF AGAIN
A055	CE 18 30		LDX	UI1	
A058	A6 01		LDAA	1,X	OFFSET FREQ
A05A	BB 91		ADDA	\$\$91	
A05C	19		DAA		
A05D	A7 01		STAA	1,X	
A05F	A6 00		LDAA	0,X	
A061	89 99		ADCA	\$\$99	
A063	19		DAA		
A064	A7 00		STAA	0,X	
A066	A6 03		LDAA	3,X	
A068	BB 95		ADDA	\$\$95	
A06A	19		DAA		
A06B	A7 03		STAA	3,X	
A06D	A6 02		LDAA	2,X	
A06F	89 99		ADCA	\$\$99	

CORRELATION COEFFICIENT

A071	19	DAA		
A072	A7 02	STAA	2,X	
A074	BD BB F4	JSR	TUNE	
A077	7D 18 56	TST	CFLAG	
A07A	27 03	BEQ	CR4	
A07C	BD A5 6E	JSR	CRDIC	
A07F	FE 18 24	CR4	LDX	COR1
A082	FF 18 86		STX	UU
A085	FE 18 26		LDX	COR3
A088	FF 18 88		STX	UU+2
A08B	5F		CLRB	
A08C	4F		CLRA	
A08D	F0 18 27		SUBB	COR4
A090	B2 18 26		SBCA	COR3
A093	B7 18 26		STAA	COR3
A096	F7 18 27		STAB	COR4
A099	BD A4 2D		JSR	CMULT
A09C	CE 18 24		LDX	*COR1
A09F	7F 18 00		CLR	TEMP1
A0A2	BD A3 F6		JSR	ACCUM
A0A5	CE 18 86		LDX	*UU
A0A8	86 20		LDAA	*32D
A0AA	B7 18 00		STAA	TEMP1
A0AD	BD A5 3A		JSR	ACCUM1
A0B0	86 01		LDAA	*1
A0B2	B7 18 3C		STAA	SENSOR
A0B5	86 04		LDAA	*4
A0B7	B7 18 35		STAA	DTIME
A0BA	BD B9 A0		JSR	DELAY1
A0BD	BD BC 0D		JSR	RDDEFT
A0C0	FE 18 24		LDX	COR1
A0C3	FF 18 8A		STX	UU
A0C6	FE 18 26		LDX	COR3
A0C9	FF 18 8C		STX	UU+2
A0CC	BD BC 0D		JSR	RDDEFT
A0CF	BD A4 D6		JSR	INDEX2
A0D2	BD BB F4		JSR	TUNE
A0D5	7D 18 56		TST	CFLAG
A0D8	27 03		BEQ	CR5
A0DA	BD A5 6E		JSR	CRDIC
A0DD	FE 18 24	CR5	LDX	COR1
A0E0	FF 18 82		STX	UV
A0E3	5F		CLRB	
A0E4	4F		CLRA	
A0E5	F0 18 27		SUBB	COR4
A0E8	B2 18 26		SBCA	COR3
A0EB	B7 18 26		STAA	COR3
A0EE	F7 18 27		STAB	COR4
A0F1	B7 18 84		STAA	UU+2
A0F4	F7 18 85		STAB	UU+3
A0F7	BD A4 2D		JSR	CMULT
A0FA	CE 18 24		LDX	*COR1
A0FD	86 08		LDAA	*8
A0FF	B7 18 00		STAA	TEMP1
A102	BD A3 F6		JSR	ACCUM
A105	CE 18 82		LDX	*UU
A108	86 28		LDAA	*40D

CORRELATION COEFFICIENT

A10A	B7 18 00	STAA	TEMP1	
A10D	BD A5 3A	JSR	ACCUM1	ALIGN AGAIN
A110	FE 18 82	LDX	UV	
A113	FF 18 24	STX	COR1	
A116	FE 18 84	LDX	UV+2	
A119	FF 18 26	STX	COR3	
A11C	FE 18 86	LDX	UV	
A11F	FF 18 8A	STX	UU	
A122	FE 18 88	LDX	UV+2	
A125	FF 18 8C	STX	UU+2	
A128	BD A4 2D	JSR	CMULT	
A12B	CE 18 24	LDX	*COR1	
A12E	86 10	LDAA	*16D	
A130	B7 18 00	STAA	TEMP1	
A133	BD A3 F6	JSR	ACCUM	CROSS: REAL
A136	CE 18 8A	LDX	*UU	
A139	86 18	LDAA	*24D	
A13B	B7 18 00	STAA	TEMP1	
A13E	BD A3 F6	JSR	ACCUM	CROSS: IMAG
A141	7D 18 57	TST	CCFLAG	
A144	26 03	BNE	CR6	
A146	7E A0 3E	JMP	CR3	
A149	CE 1E 44	CR6	LDX #ARRAY+4	FINISHED SCANNING
A14C	4F	CLRA		
A14D	BD BB 8E	JSR	PUSH44	
A150	86 05	LDAA	#5	
A152	B7 18 35	STAA	DTIME	
A155	CE 10 04	LDX	#\$1004	
A158	FF 18 3D	STX	NSAMP	
A15B	CE 1E 40	LDX	#ARRAY	
A15E	4F	CLRA		
A15F	BD BB 8E	JSR	PUSH44	
A162	7F 18 00	CLR	TEMP1	
A165	BD A5 9C	JSR	MEAN	
A168	CE 1E 4C	LDX	#ARRAY+12D	
A16B	4F	CLRA		
A16C	BD BB 8E	JSR	PUSH44	
A16F	CE 1E 48	LDX	#ARRAY+8	
A172	4F	CLRA		
A173	BD BB 8E	JSR	PUSH44	
A176	86 08	LDAA	#8	
A178	B7 18 00	STAA	TEMP1	
A17B	BD A5 9C	JSR	MEAN	
A17E	86 01	LDAA	#1	
A180	BD 80 00	JSR	MATH	DENOMINATOR SQUARED
A183	CE 18 7A	LDX	#VP	FIND SQUARE ROOT
A186	4F	CLRA		
A187	BD BB 68	JSR	PULL4	
A18A	CE 18 7E	LDX	#UP	
A18D	4F	CLRA		
A18E	BD BB 68	JSR	PULL4	VP, UP=DEN SQ
A191	FE 18 7A	LDX	VP	COPY
A194	FF 18 82	STX	UV	
A197	FE 18 7C	LDX	VP+2	
A19A	FF 18 84	STX	UV+2	
A19D	FE 18 7E	LDX	UP	
A1A0	FF 18 86	STX	UV	

CORRELATION COEFFICIENT

A1A3	FE 18 80	LDX	UP+2	
A1A6	FF 18 88	STX	VV+2	
A1A9	B6 18 89	LDAA	VV+3	EXP
A1AC	47	ASRA		
A1AD	B7 18 89	STAA	VV+3	UV,UV=INITIAL ITERATE
A1B0	86 0A	LDAA	\$10D	
A1B2	B7 18 00	STAA	TEMP1	
A1B5	CE 18 7E	LDX	\$UP	ITERATE
A1B8	4F	CLRA		
A1B9	BD BB 8E	JSR	PUSH44	
A1BC	CE 18 7A	LDX	#VP	
A1BF	4F	CLRA		
A1C0	BD BB 8E	JSR	PUSH44	
A1C3	CE 18 86	LDX	#VV	
A1C6	4F	CLRA		
A1C7	BD BB 8E	JSR	PUSH44	
A1CA	CE 18 82	LDX	#UV	
A1CD	4F	CLRA		
A1CE	BD BB 8E	JSR	PUSH44	
A1D1	86 02	LDAA	#2	
A1D3	BD 80 00	JSR	MATH	X=(DEN SQ)/ITERATE
A1D6	CE 18 86	LDX	#VV	
A1D9	4F	CLRA		
A1DA	BD BB 8E	JSR	PUSH44	
A1DD	CE 18 82	LDX	#UV	
A1E0	4F	CLRA		
A1E1	BD BB 8E	JSR	PUSH44	
A1E4	86 06	LDAA	#6	
A1E6	BD 80 00	JSR	MATH	X=X+ITERATE
A1E9	30	TSX		
A1EA	6A 07	DEC	7,X	X=0.5X
A1EC	CE 18 82	LDX	#UV	
A1EF	4F	CLRA		
A1F0	BD BB 68	JSR	PULL4	
A1F3	CE 18 86	LDX	#VV	
A1F6	4F	CLRA		
A1F7	BD BB 68	JSR	PULL4	UV,UV=NEW ITERATE
A1FA	7A 18 00	DEC	TEMP1	
A1FD	26 B6	BNE	CR8	
A1FF	CE 1E 54	LDX	\$ARRAY+20D	SQUARE ROOT DONE
A202	4F	CLRA		
A203	BD BB 8E	JSR	PUSH44	
A206	CE 1E 50	LDX	\$ARRAY+16D	
A209	4F	CLRA		
A20A	BD BB 8E	JSR	PUSH44	CROSS: REAL
A20D	4F	CLRA		
A20E	BD BB B4	JSR	PUSH41	
A211	CE 1E 60	LDX	\$ARRAY+32D	
A214	4F	CLRA		
A215	BD BB 8E	JSR	PUSH44	A
A218	86 07	LDAA	#7	
A21A	BD 80 00	JSR	MATH	
A21D	4F	CLRA		
A21E	BD BB B4	JSR	PUSH41	
A221	CE 1E 68	LDX	\$ARRAY+40D	
A224	4F	CLRA		
A225	BD BB 8E	JSR	PUSH44	C

CORRELATION COEFFICIENT

A228	86 07	LDAA	\$7	
A22A	BD 80 00	JSR	MATH	
A22D	86 01	LDAA	\$1	
A22F	BD 80 00	JSR	MATH	AC
A232	4F	CLRA		
A233	BD BB B4	JSR	PUSH41	
A236	CE 1E 64	LDX	\$ARRAY+36D	
A239	4F	CLRA		
A23A	BD BB BE	JSR	PUSH44	B
A23D	86 07	LDAA	\$7	
A23F	BD 80 00	JSR	MATH	
A242	4F	CLRA		
A243	BD BB B4	JSR	PUSH41	
A246	CE 1E 6C	LDX	\$ARRAY+44D	
A249	4F	CLRA		
A24A	BD BB BE	JSR	PUSH44	D
A24D	86 07	LDAA	\$7	
A24F	BD 80 00	JSR	MATH	
A252	86 01	LDAA	\$1	
A254	BD 80 00	JSR	MATH	BD
A257	86 05	LDAA	\$5	
A259	BD 80 00	JSR	MATH	AC-BD
A25C	86 05	LDAA	\$5	
A25E	BD 80 00	JSR	MATH	
A261	CE 18 86	LDX	\$UV	
A264	4F	CLRA		
A265	BD BB BE	JSR	PUSH44	
A268	CE 18 82	LDX	\$UV	
A26B	4F	CLRA		
A26C	BD BB BE	JSR	PUSH44	
A26F	86 02	LDAA	\$2	
A271	BD 80 00	JSR	MATH	REAL/DEN
A274	CE 18 7A	LDX	\$VP	
A277	4F	CLRA		
A278	BD BB 68	JSR	PULL4	
A278	CE 18 7E	LDX	\$UP	
A27E	4F	CLRA		
A27F	BD BB 68	JSR	PULL4	VP, UP=REAL FPN
A282	CE 1E 5C	LDX	\$ARRAY+28D	
A285	4F	CLRA		
A286	BD BB BE	JSR	PUSH44	
A289	CE 1E 58	LDX	\$ARRAY+24D	
A28C	4F	CLRA		
A28D	BD BB BE	JSR	PUSH44	CROSS: IMAG
A290	4F	CLRA		
A291	BD BB B4	JSR	PUSH41	
A294	CE 1E 60	LDX	\$ARRAY+32D	
A297	4F	CLRA		
A298	BD BB BE	JSR	PUSH44	A
A29B	86 07	LDAA	\$7	
A29D	BD 80 00	JSR	MATH	
A2A0	4F	CLRA		
A2A1	BD BB B4	JSR	PUSH41	
A2A4	CE 1E 6C	LDX	\$ARRAY+44D	
A2A7	4F	CLRA		
A2A8	BD BB BE	JSR	PUSH44	D
A2A8	86 07	LDAA	\$7	

CORRELATION COEFFICIENT

A2AD	BD 80 00	JSR	MATH	
A2B0	86 01	LDAA	#1	
A2B2	BD 80 00	JSR	MATH	AD
A2B5	4F	CLRA		
A2B6	BD BB B4	JSR	PUSH41	
A2B9	CE 1E 64	LDX	#ARRAY+36D	
A2BC	4F	CLRA		
A2BD	BD BB 8E	JSR	PUSH44	B
A2C0	86 07	LDAA	#7	
A2C2	BD 80 00	JSR	MATH	
A2C5	4F	CLRA		
A2C6	BD BB B4	JSR	PUSH41	
A2C9	CE 1E 68	LDX	#ARRAY+40D	
A2CC	4F	CLRA		
A2CD	BD BB 8E	JSR	PUSH44	C
A2D0	86 07	LDAA	#7	
A2D2	BD 80 00	JSR	MATH	
A2D5	86 01	LDAA	#1	
A2D7	BD 80 00	JSR	MATH	BC
A2DA	86 06	LDAA	#6	
A2DC	BD 80 00	JSR	MATH	AD+BC
A2DF	86 05	LDAA	#5	
A2E1	BD 80 00	JSR	MATH	
A2E4	CE 18 86	LDX	#UV	
A2E7	4F	CLRA		
A2E8	BD BB 8E	JSR	PUSH44	
A2EB	CE 18 82	LDX	#UV	
A2EE	4F	CLRA		
A2EF	BD BB 8E	JSR	PUSH44	
A2F2	86 02	LDAA	#2	
A2F4	BD 80 00	JSR	MATH	IMAG/DEN
A2F7	CE 18 82	LDX	#UV	
A2FA	4F	CLRA		
A2FB	BD BB 68	JSR	PULL4	
A2FE	CE 18 86	LDX	#UV	
A301	4F	CLRA		
A302	BD BB 68	JSR	PULL4	UV,UV=IMAG FPN
A305	F6 18 81	LDAB	UP+3	FLOAT TO FIX REAL
A308	50	NEGB		
A309	CE 18 7A	LDX	#UP	
A30C	5D	CR9	TSTB	
A30D	27 0D	BEQ	CR10	
A30F	A6 00	LDAA	0,X	
A311	47	ASRA		
A312	A7 00	STAA	0,X	
A314	A6 01	LDAA	1,X	
A316	46	RORA		
A317	A7 01	STAA	1,X	
A319	5A	DEC8		
A31A	20 F0	BRA	CR9	
A31C	F6 18 89	CR10	LDAB	UV+3
A31F	50	NEGB		FLOAT TO FIX IMAG
A320	CE 18 82	LDX	#UV	
A323	5D	CR11	TSTB	
A324	27 0D	BEQ	CR12	
A326	A6 00	LDAA	0,X	
A328	47	ASRA		

CORRELATION COEFFICIENT

A329	A7 00	STAA	0,X	
A32B	A6 01	LDAA	1,X	
A32D	46	RORA		
A32E	A7 01	STAA	1,X	
A330	5A	DEC8		
A331	20 F0	BRA	CR11	
A333	FE 18 7A	CR12	LDX	VP
A336	FF 18 24	STX	COR1	
A339	FE 18 82	LDX	UV	
A33C	FF 18 26	STX	COR3	
A33F	7F 18 2A	CLR	COR9	
A342	7F 18 2B	CLR	COR10	
A345	BD BD D6	JSR	CORDIC	
A348	FE 18 24	LDX	COR1	
A34B	FF 18 00	STX	TEMP1	
A34E	7F 18 02	CLR	TEMP3	
A351	7F 18 03	CLR	TEMP4	
A354	CE 18 00	LDX	\$TEMP1	
A357	4F	CLRA		
A358	BD BB 8E	JSR	PUSH44	
A35B	86 5A	LDAA	\$\$5A	
A35D	36	PSHA		
A35E	86 59	LDAA	\$\$59	
A360	36	PSHA		
A361	86 01	LDAA	\$1	
A363	36	PSHA		
A364	4A	DECA		
A365	36	PSHA	SF=.41169*65536*32768/10000	
A366	86 09	LDAA	\$9	
A368	BD 80 00	JSR	MATH	
A36B	CE 18 00	LDX	SCALE MAG	
A36E	4F	CLRA		
A36F	BD BB 68	JSR	PULL4	
A372	BD BB 17	JSR	BI\$BCD	
A375	FE 18 06	LDX	BCD1	
A378	FF 18 24	STX	COR1	
A378	FE 18 2A	LDX	MAG IN BCD	
A37E	FF 18 00	STX	TEMP1	
A381	7F 18 02	CLR	TEMP3	
A384	7F 18 03	CLR	TEMP4	
A387	CE 18 00	LDX	\$TEMP1	
A38A	4F	CLRA		
A38B	BD BB 8E	JSR	PUSH44	
A38E	86 56	LDAA	\$\$56	
A390	36	PSHA		
A391	86 34	LDAA	\$\$34	
A393	36	PSHA		
A394	86 12	LDAA	\$\$12	
A396	36	PSHA		
A397	4F	CLRA		
A398	36	PSHA	SF=32768*65536/1800	
A399	86 09	LDAA	\$9	
A39B	BD 80 00	JSR	MATH	
A39E	CE 18 00	LDX	SCALE PHASE	
A3A1	4F	CLRA		
A3A2	BD BB 68	JSR	PULL4	
A3A5	B6 18 00	LDAA	TEMP1	

CORRELATION COEFFICIENT

A3AB	B7 18 2A	STAA	COR9
A3AB	2A 0E	BPL	CR13
A3AD	4F	CLRA	
A3AE	5F	CLRB	
A3AF	F0 18 03	SUBB	TEMP4
A3B2	B2 18 02	SBCA	TEMP3
A3B5	B7 18 02	STAA	TEMP3
A3B8	F7 18 03	STAB	TEMP4
A3BB	BD BB 17	JSR	BI\$BCD
A3BE	FE 18 06	LDX	BCD1
A3C1	FF 18 26	STX	COR3
A3C4	7D 18 2A	TST	COR9
A3C7	2A 01	BPL	CORDSP
A3C9	0D	SEC	
A3CA	07	CORDSP	TPA
A3CB	36	PSHA	SAVE SIGN
A3CC	86 0A	LDAA	\$10
A3CE	B7 ED 30	STAA	DISPTR+1
A3D1	CE 18 26	LDX	*COR1+2
A3D4	86 10	LDAA	\$\$10
A3D6	8D 16	BSR	CORDS1
A3D8	86 03	LDAA	3
A3DA	B7 ED 30	STAA	DISPTR+1
A3DD	CE 18 24	LDX	*COR1
A3E0	86 FF	LDAA	\$\$FF
A3E2	8D 0A	BSR	CORDS1
A3E4	86 01	LDAA	\$1
A3E6	B7 ED 03	STAA	LEDBFR+3
A3E9	32	FULA	
A3EA	06	TAP	
A3EB	7E AE F7	JMP	CORCRT
A3EE	36	CORDS1	PSHA
A3EF	A6 00	LDAA	0,X
A3F1	E6 01	LDAB	1,X
A3F3	7E AB 66	G1	JMP
	*		FIXSGN
	*		
	*		SUBROUTINE ACCUM
	*		ACCUMULATES PARTIAL PRODUCTS
A3F6	4F	ACCUM	CLRA
A3F7	BD BB B4	JSR	PUSH41
A3FA	4F	CLRA	
A3FB	BD BB 8E	JSR	PUSH44
A3FE	86 07	LDAA	\$7
A400	BD 80 00	JSR	MATH
A403	CE 1E 44	LDX	*ARRAY+4
A406	B6 18 00	LDAA	TEMP1
A409	BD BB 8E	JSR	PUSH44
A40C	CE 1E 40	LDX	*ARRAY
A40F	B6 18 00	LDAA	TEMP1
A412	BD BB 8E	JSR	PUSH44
A415	86 06	LDAA	\$6
A417	BD 80 00	JSR	MATH
A41A	CE 1E 40	LDX	*ARRAY
A41D	B6 18 00	LDAA	TEMP1
A420	BD BB 68	JSR	PULL4
A423	CE 1E 44	LDX	*ARRAY+4

CORRELATION COEFFICIENT

A426	B6	18 00	LDAA	TEMP1	
A429	BD	BB 68	JSR	PULL4	
A42C	39		RTS		
	*				
	*				
	*				
					SUBROUTINE CMULT
	*				DOES COMPLEX MULTIPLY
	*				(COR1,COR2+JCOR3,COR4)*(UU,UU+1+JUU+2,UU+3)
	*				RETURNS REAL IN COR1 TO COR4; IMAG IN UU TO UU+3
A42D	F6	18 25	CMULT	LDAB	COR2
A430	B6	18 24		LDAA	COR1
A433	BD	BB D1		JSR	PUSH42
A436	F6	18 27		LDAB	COR4
A439	B6	18 26		LDAA	COR3
A43C	BD	BB D1		JSR	PUSH42
A43F	B6	0C		LDAA	#12D
A441	BD	80 00		JSR	MATH
A444	F6	18 8B		LDAB	UU+1
A447	B6	18 8A		LDAA	UU
A44A	BD	BB D1		JSR	PUSH42
A44D	F6	18 8D		LDAB	UU+3
A450	B6	18 8C		LDAA	UU+2
A453	BD	BB D1		JSR	PUSH42
A456	B6	0B		LDAA	#11D
A458	BD	80 00		JSR	MATH
A45B	B6	0A		LDAA	#10D
A45D	BD	80 00		JSR	MATH
					(A-B)*(C+D)
A460	F6	18 25		LDAB	COR2
A463	B6	18 24		LDAA	COR1
A466	BD	BB D1		JSR	PUSH42
A469	F6	18 8D		LDAB	UU+3
A46C	B6	18 8C		LDAA	UU+2
A46F	BD	BB D1		JSR	PUSH42
A472	B6	0A		LDAA	#10D
A474	BD	80 00		JSR	MATH
A477	CE	18 7A		LDX	#VP
A47A	4F			CLRA	
A47B	BD	BB 68		JSR	PULL4
A47E	F6	18 27		LDAB	COR4
A481	B6	18 26		LDAA	COR3
A484	BD	BB D1		JSR	PUSH42
A487	F6	18 8B		LDAB	UU+1
A48A	B6	18 8A		LDAA	UU
A48D	BD	BB D1		JSR	PUSH42
A490	B6	0A		LDAA	#10D
A492	BD	80 00		JSR	MATH
A495	CE	18 7E		LDX	#UP
A498	4F			CLRA	
A499	BD	BB 68		JSR	PULL4
A49C	CE	18 7E		LDX	#UP
A49F	4F			CLRA	
A4A0	BD	BB 8E		JSR	PUSH44
A4A3	CE	18 7A		LDX	#VP
A4A6	4F			CLRA	
A4A7	BD	BB 8E		JSR	PUSH44
A4AA	B6	0B		LDAA	#11D
A4AC	BD	80 00		JSR	MATH
					IMAG=AD+BC

CORRELATION COEFFICIENT

A4AF	CE 18 8A	LDX	\$UU	
A4B2	4F	CLRA		
A4B3	BD BB 68	JSR	PULL4	
A4B6	CE 18 7E	LDX	\$UP	
A4B9	4F	CLRA		
A4BA	BD BB 8E	JSR	PUSH44	
A4BD	CE 18 7A	LDX	\$VP	
A4C0	4F	CLRA		
A4C1	BD BB 8E	JSR	PUSH44	
A4C4	86 0C	LDAA	\$12D	
A4C6	BD 80 00	JSR	MATH	BC-AD
A4C9	86 0B	LDAA	\$11D	
A4CB	BD 80 00	JSR	MATH	REAL=(A-B)*(C+D)+(BC-AD)
A4CE	CE 18 24	LDX	\$COR1	
A4D1	4F	CLRA		
A4D2	BD BB 68	JSR	PULL4	
A4D5	39	RTS		

*

*

*

SUBROUTINE INDEX2
INCREMENTS FREQUENCIES

				OFFSET FREQ
A4D6	CE 18 30	INDEX2	LDX	\$UI1
A4D9	A6 01		LDAA	1,X
A4DB	BB 09		ADDA	\$9
A4DD	19		DAA	
A4DE	A7 01		STAA	1,X
A4E0	A6 00		LDAA	0,X
A4E2	89 00		ADCA	\$0
A4E4	19		DAA	
A4E5	A7 00		STAA	0,X
A4E7	A6 03		LDAA	3,X
A4E9	BB 05		ADDA	\$5
A4EB	19		DAA	
A4EC	A7 03		STAA	3,X
A4EE	A6 02		LDAA	2,X
A4F0	89 00		ADCA	\$0
A4F2	19		DAA	
A4F3	A7 02		STAA	2,X
A4F5	A6 01		LDAA	1,X
A4F7	BB 90		ADDA	\$\$90
A4F9	19		DAA	
A4FA	A7 01		STAA	1,X
A4FC	A6 00		LDAA	0,X
A4FE	89 99		ADCA	\$\$99
A500	19		DAA	
A501	A7 00		STAA	0,X
A503	A6 01		LDAA	1,X
A505	BB 74		ADDA	\$\$74
A507	19		DAA	
A508	A6 00		LDAA	0,X
A50A	89 66		ADCA	\$\$66
A50C	19		DAA	
A50D	81 99		CMPA	\$\$99
A50F	27 01		BEQ	IEX1
A511	39		RTS	
A512	CE 39 26	IEX1	LDX	\$\$3926
A515	FF 18 30		STX	UI1

CORRELATION COEFFICIENT

A518	CE 18 30	LDX	\$UI1	
A51B	A6 03	LDAA	3,X	INC VJ 100 KHZ
A51D	8B 10	ADDA	\$\$10	
A51F	19	IAA		
A520	A7 03	STAA	3,X	
A522	A6 02	LDAA	2,X	
A524	89 00	ADCA	0	
A526	19	DA		
A527	A7 02	STAA	2,X	
A529	A6 03	LDAA	3,X	
A52B	8B 38	ADDA	\$\$38	
A52D	19	IAA		
A52E	A6 02	LDAA	2,X	
A530	89 63	ADCA	\$\$63	
A532	19	DA		
A533	27 01	BEQ	IEX2	
A535	39	RTS		
A536	7C 18 57	IEX2	INC	CCFLAG
A539	39	RTS		
	*			
	*			
A53A	FF 18 01	ACCUM1	STX	TEMP2
A53D	8D 10	BSR	ACCUM2	
A53F	B6 18 00	LDAA	TEMP1	
A542	8B 04	ADDA	4	
A544	B7 18 00	STAA	TEMP1	
A547	FE 18 01	LDX	TEMP2	
A54A	08	INX		
A54B	08	INX		
A54C	8D 01	BSR	ACCUM2	
A54E	39	RTS		
	*			
A54F	E6 01	ACCUM2	LDAB	1,X
A551	A6 00	LDAA	0,X	
A553	BD BB D1	JSR	PUSH42	
A556	CE 1E 40	LDX	ARRAY	
A559	B6 18 00	LDAA	TEMP1	
A55C	BD BB 8E	JSR	PUSH44	
A55F	86 0B	LDAA	11D	
A561	BD 80 00	JSR	MATH	ACCUM
A564	CE 1E 40	LDX	ARRAY	
A567	B6 18 00	LDAA	TEMP1	
A56A	BD BB 68	JSR	PULL4	
A56D	39	RTS		
	*			
	*			
A56E	7F 18 2A	CRDIC	CLR	COR9
A571	7F 18 2B	CLR		COR10
A574	BD BD D6	JSR	CORDIC	
A577	FE 18 8C	LDX	UU+2	
A57A	FF 18 26	STX	COR3	
A57D	FE 18 24	LDX	COR1	
A580	FF 18 00	STX	TEMP1	
A583	FE 18 8A	LDX	UU	
A586	FF 18 24	STX	COR1	
A589	FE 18 00	LDX	TEMP1	
A58C	FF 18 8A	STX	UU	

CORRELATION COEFFICIENT

A58F	CE 00 00	LDX	\$0	
A592	FF 18 8C	STX	UU+2	
A595	FF 18 2A	STX	COR9	
A598	BD BD D6	JSR	CORDIC	
A59B	39	RTS		
	*			
	*			
A59C	86 01	MEAN	LDAA	#1
A59E	B7 18 04		STAA	TEMP5
A5A1	32		PULA	
A5A2	B7 18 02		STAA	TEMP3
A5A5	32		PULA	
A5A6	B7 18 03		STAA	TEMP4
A5A9	4F	M1	CLRA	
A5AA	BD BB B4		JSR	PUSH41
A5AD	CE 1E 60		LDX	#ARRAY+32D
A5B0	B6 18 00		LDAA	TEMP1
A5B3	BD BB BE		JSR	PUSH44
A5B6	86 07		LDAA	#7
A5B8	BD 80 00		JSR	MATH
A5BB	7A 18 04		DEC	TEMP5
A5BE	2A E9		BPL	M1
A5C0	86 01		LDAA	#1
A5C2	B7 18 04		STAA	TEMP5
A5C5	BD 80 00		JSR	MATH
A5C8	4F	M2	CLRA	
A5C9	BD BB B4		JSR	PUSH41
A5CC	CE 1E 64		LDX	#ARRAY+36D
A5CF	B6 18 00		LDAA	TEMP1
A5D2	BD BB BE		JSR	PUSH44
A5D5	86 07		LDAA	#7
A5D7	BD 80 00		JSR	MATH
A5DA	7A 18 04		DEC	TEMP5
A5DD	2A E9		BPL	M2
A5DF	86 01		LDAA	#1
A5E1	BD 80 00		JSR	MATH
A5E4	86 06		LDAA	#6
A5E6	BD 80 00		JSR	MATH
A5E9	86 05		LDAA	#5
A5EB	BD 80 00		JSR	MATH
A5EE	B6 18 03		LDAA	TEMP4
A5F1	36		PSHA	
A5F2	B6 18 02		LDAA	TEMP3
A5F5	36		PSHA	
A5F6	39		RTS	
A5F7			END	

STATEMENTS =731

FREE BYTES =489

NO ERRORS DETECTED

MORE MEASURE SUBROUTINES

1800		ORG	\$1800
1800	TEMP1	RMB	1
1801	TEMP2	RMB	1
1802	TEMP3	RMB	1
1803	TEMP4	RMB	1
1804	TEMP5	RMB	1
1805	TEMP6	RMB	1
1806	BCD1	RMB	1
1807	BCD2	RMB	1
1808	CTR	RMB	1
1809	CTR1	RMB	1
1812		ORG	\$1812
1812	PHASE	RMB	2
1814	XZERO	RMB	4
1818	YZERO	RMB	4
181C	X1ZERO	RMB	4
1820	Y1ZERO	RMB	4
1824	COR1	RMB	1
1825	COR2	RMB	1
1826	COR3	RMB	1
1827	COR4	RMB	3
182A	COR9	RMB	1
182B	COR10	RMB	1
182C	I	RMB	1
182D	J	RMB	1
182E	ID	RMB	1
182F	JD	RMB	1
1830	UI1	RMB	1
1831	UI2	RMB	1
1832	VJ1	RMB	1
1833	VJ2	RMB	1
1834	MODE	RMB	1
1835	DTIME	RMB	1
1836	JSTART	RMB	1
1837	BDRY	RMB	1
1838	BDRYID	RMB	1
1839	NBDRY	RMB	1
183A	BDRYFT	RMB	1
183B	FIRST	RMB	1
183C	SENSOR	RMB	1
183D	NSAMP	RMB	1
183E	LOGNS	RMB	1
183F	BCDPTR	RMB	2
1841	L	RMB	1
1842	LIMIT	RMB	1
1843	X	RMB	1
1844	Y	RMB	1
1845	ZERO	RMB	4
1849	X1INC	RMB	2
184B	Y1INC	RMB	2
184D	NPT	RMB	1
184E	STACK1	RMB	1
184F	STACK2	RMB	1
1850	PUSHST	RMB	1
1851	THR11	RMB	1
1852	THR12	RMB	1
1853	THR21	RMB	1

MORE MEASURE SUBROUTINES

1854	THR22	RMB	1
1855	HOW	RMB	1
1856	IJPTR	RMB	1
1857	IJFTR1	RMB	1
1858	ADRSW	RMB	1
1859	SETUP	RMB	1
185A	FINE	RMB	1
185B	K\$G	RMB	1
185C	KKKK	RMB	4
1860	CORSE	RMB	2
1862	ST	RMB	2
1864	SAVESF	RMB	2
1866	MAX	RMB	4
186A	SUM	RMB	4
186E	XTRAN	RMB	2
1870	YTRAN	RMB	2
1872	DELX	RMB	4
1876	DELY	RMB	4
187A	UP	RMB	4
187E	UP	RMB	4
1882	UV	RMB	4
1886	UU	RMB	4
188A	STEPS	RMB	2
1890	ANGLE	RMB	2
1892	SMAG	RMB	4
1896	SAVEX	RMB	2
1898	SAVEY	RMB	2
189A	DIRECT	RMB	1
189B	DTHR11	RMB	1
189C	DTHR12	RMB	1
189D	DTHR21	RMB	1
189E	DTHR22	RMB	1
189F	MPLX	RMB	2
18A1	SIGNI	RMB	1
18A2	SIGNQ	RMB	1
18A3	SFLAG	RMB	1
18A4	SNFLAG	RMB	1
18D8	ORG		\$18D8
18D8	DELTA	RMB	10D
18E2	BCDAR	RMB	512D
1AE2	BDRYLF	RMB	5
1AE7	BDRYCT	RMB	5
1AEC	S1VP1M	RMB	20D
1B00	S1UP1M	RMB	20D
1B14	SIUV1M	RMB	20D
1B28	S1V1M	RMB	20D
1B3C	S1U1M	RMB	20D
1B50	IJ	RMB	400D
1CEO	PS1MS	RMB	400D
1E70	MS1MS	RMB	400D
8000	MATH	EQU	\$8000
BDD6	CORDIC	EQU	\$BDD6
AE2A	SYNDET	EQU	\$AE2A
BBD1	PUSH42	EQU	\$BBB1
BB68	PULL4	EQU	\$BB68
BCCF	INDEX	EQU	\$BCCF

MORE MEASURE SUBROUTINES

B9B6		ADDRES	EQU	\$B9B6
BBF4		TUNE	EQU	\$BBF4
B9A0		DELAY1	EQU	\$B9A0
BC0D		RDDEFT	EQU	\$BC0D
A5F7			ORG	\$A5F7
		*		
A5F7	BD BD D6	CORD	JSR	CORDIC
A5FA	FE 18 2A		LDX	COR9
A5FD	FF 18 00		STX	TEMP1
A600	FE 18 12		LDX	PHASE
A603	FF 18 02		STX	TEMP3
A606	CE 18 00		LDX	*TEMP1
A609	BD A7 4C		JSR	PHASDF
A60C	EE 00		LDX	0,X
A60E	FF 18 2A		STX	COR9
A611	39		RTS	
		*		
A612	08	IXST	INX	
A613	08		INX	
A614	08		INX	
A615	FF 18 3F		STX	BCDFTR
A618	39		RTS	
		*		
A619	B6 18 9F	CALSDI	LDAA	MPLX
A61C	B7 DF F9		STAA	\$DFF9
A61F	CE 18 00		LDX	*TEMP1
A622	BD AE 2A		JSR	SYNDET
A625	7D 18 00		TST	TEMP1
A628	BD 0C		BSR	SGNSET
A62A	B7 18 A1		STAA	SIGNI
A62D	7D 18 02		TST	TEMP3
A630	BD 04		BSR	SGNSET
A632	B7 18 A2		STAA	SIGNQ
A635	39		RTS	
		*		
A636	2A 03	SGNSET	BPL	SETPOS
A638	86 FF		LDAA	#\$FF
A63A	39		RTS	
A63B	4F	SETPOS	CLRA	
A63C	39		RTS	
		*		SUBROUTINE INDEX
		*		INDEXES ARRAYS
A63D	FF 18 00	INDEX	STX	TEMP1
A640	B7 18 02		STAA	TEMP3
A643	F7 18 03		STAB	TEMP4
A646	B6 18 00		LDAA	TEMP1
A649	F6 18 01		LDAB	TEMP2
A64C	FB 18 03		ADDB	TEMP4
A64F	B9 18 02		ADCA	TEMP3
A652	B7 18 00		STAA	TEMP1
A655	F7 18 01		STAB	TEMP2
A658	FE 18 00		LDX	TEMP1
A65B	39		RTS	
		*		SUBROUTINE INDEX1
		*		EXTENDS SEARCH OUT WX-AXIS
A65C	BD BC CF	INDEX1	JSR	INDEX
A65F	B6 18 2D		LDAA	J

MORE MEASURE SUBROUTINES

A662	81 77		CMPA	\$119D
A664	2E 06		BGT	IND1
A666	40		NEGA	
A667	81 77		CMPA	\$119D
A669	2E 0C		BGT	IND2
A66B	39		RTS	
A66C	86 77	IND1	LDAA	\$119D
A66E	B7 18 2D		STAA	J
A671	8B 07		ADDA	\$7
A673	B7 18 2C		STAA	I
A676	39		RTS	
A677	86 77	IND2	LDAA	\$119D
A679	B7 18 2D		STAA	J
A67C	B6 18 2C		LDAA	I
A67F	8B 07		ADDA	\$7
A681	B7 18 2C		STAA	I
A684	39		RTS	
*				
SUBROUTINE UPDATE				
*				
A685	CE 00 00	UPDATE	LDX	\$0
A688	FF 18 51		STX	THR11
A68B	FF 18 43		STX	X
A68E	F6 18 25	UP1	LDAB	COR2
A691	B6 18 24		LDAA	COR1
A694	F0 18 52		SUBB	THR12
A697	B2 18 51		SBCA	THR11
A69A	2D 34		BLT	UP2
A69C	CE 1B 50		LDX	\$IJ
				MAG>THR(X)
A69F	F6 18 44		LDAB	Y
A6A2	B6 18 43		LDAA	X
A6A5	BD A6 3D		JSR	INDEX
A6A8	E6 00		LDAB	0,X
A6AA	A6 01		LDAA	1,X
A6AC	CB 01		ADDB	\$1
A6AE	89 00		ADCA	\$0
A6B0	E7 00		STAB	0,X
A6B2	A7 01		STAA	1,X
A6B4	F6 18 52		LDAB	THR12
A6B7	B6 18 51		LDAA	THR11
A6BA	FB 18 54		ADDB	THR22
A6BD	B9 18 53		ADCA	THR21
A6C0	F7 18 52		STAB	THR12
A6C3	B7 18 51		STAA	THR11
A6C6	FE 18 43		LDX	X
A6C9	08		INX	
A6CA	08		INX	
A6CB	FF 18 43		STX	X
A6CE	20 BE		BRA	UP1
A6D0	39	UP2	RTS	THR>MAG
*				
SUBROUTINE THRSET				
*				
SETS THRESHOLD				
A6D1	7F 18 43	THRSET	CLR	X
A6D4	7F 18 44		CLR	Y
A6D7	CE 7F FF		LDX	\$\$7FFF
A6DA	FF 18 66		STX	MAX
A6DD	CE 1B 50	TH1	LDX	\$IJ
A6E0	F6 18 44		LDAB	Y

MORE MEASURE SUBROUTINES

A6E3	B6 18 43	LDAA	X	
A6E6	BD A6 3D	JSR	IDEX	
A6E9	E6 00	LDAB	0,X	N(X)
A6EB	A6 01	LDAA	1,X	
A6ED	5D	TSTB		
A6EE	26 03	BNE	TH2	
A6F0	4D	TSTA		
A6F1	27 32	BEQ	TH6	
A6F3	F0 18 4D	SUBB	NPT	N>0
A6F6	82 00	SBCA	#0	
A6F8	2C 07	BGE	TH4	
A6FA	50	NEGB		
A6FB	25 03	BCS	TH3	
A6FD	40	NEGA		
A6FE	20 01	BRA	TH4	
A700	43	TH3	COMA	
A701	F7 18 69	TH4	STAB	MAX+3
A704	B7 18 68	STAA	MAX+2	ABS(N-NPT)
A707	F0 18 67	SUBB	MAX+1	
A70A	B2 18 66	SBCA	MAX	
A70D	2C 0C	BGE	TH5	
A70F	FE 18 68	LDX	MAX+2	
A712	FF 18 66	STX	MAX	
A715	FE 18 43	LDX	X	
A718	FF 18 8A	STX	UU	
A71B	FE 18 43	TH5	LDX	X
A71E	08	INX		
A71F	08	INX		
A720	FF 18 43	STX	X	
A723	20 B8	BRA	TH1	
A725	F6 18 8B	TH6	LDAB	UU+1
A728	B6 18 8A	LDAA	UU	N=0
A72B	47	ASRA		
A72C	56	RORB		
A72D	BD BB D1	JSR	PUSH42	
A730	F6 18 54	LDAB	THR22	
A733	B6 18 53	LDAA	THR21	
A736	BD BB D1	JSR	PUSH42	
A739	86 0A	LDAA	#10D	
A73B	BD B0 00	JSR	MATH	XXTHR2
A73E	CE 18 8A	LDX	#UU	
A741	4F	CLRA		
A742	BD BB 68	JSR	PULL4	
A745	FE 18 8C	LDX	UU+2	
A748	FF 18 51	STX	THR11	THRESHOLD SET
A74B	39	RTS		
	*	SUBROUTINE PHASDF		
	*	COMPUTES PHASE1-PHASE2		
	*	ENTER WITH PHASE1,PHASE2 IN ADJACENT MEMORY		
	*	X POINTS TO PHASE1		
	*	PHASE DIFF RETURNED IN PLACE OF PHASE1		
A74C	A6 00	PHASDF	LDAA	0,X
A74E	E6 01		LDAB	1,X
A750	E0 03		SUBB	3,X
A752	A2 02		SBCA	2,X
A754	28 24		BVC	PDF2
A756	A6 00		LDAA	0,X
				OVERFLOW

MORE MEASURE SUBROUTINES

A758	E6 01		LDAB	1,X	
A75A	CB FF		ADDB	#\$FF	P1+PHI
A75C	89 7F		ADCA	#\$7F	
A75E	29 0A		BVS	PDF1	
A760	E0 03		SUBB	3,X	
A762	A2 02		SBCA	2,X	
A764	CB FF		ADDB	#\$FF	PD=(P1+PHI)-(P2-PHI)
A766	89 7F		ADCA	#\$7F	
A768	20 10		BRA	PDF2	
A76A	A6 00	PDF1	LDAA	0,X	P1+PHI>PHI
A76C	E6 01		LDAB	1,X	
A76E	C0 FF		SUBB	#\$FF	
A770	82 7F		SBCA	#\$7F	
A772	E0 03		SUBB	3,X	
A774	A2 02		SBCA	2,X	
A776	C0 FF		SUBB	#\$FF	PD=(P1-PHI)-(P2+PHI)
A778	82 7F		SBCA	#\$7F	
A77A	A7 00	PDF2	STAA	0,X	PD
A77C	E7 01		STAB	1,X	
A77E	39		RTS		

* SUBROUTINE PHSSET

* MEASURES PHASE DIFF AT ORIGINS AND STORES IN PHASE

A77F	7D 18 A4	PHSSET	TST	SNFLAG
A782	2E 65		BGT	PSS
A784	7F 18 58		CLR	ADRSW
A787	7F 18 2C		CLR	I
A78A	7F 18 2D		CLR	J
A78D	7F 18 3C		CLR	SENSOR
A790	BD B9 B6		JSR	ADDRES
A793	BD BB F4		JSR	TUNE
A796	7A 18 58		DEC	ADRSW
A799	7A 18 3C		DEC	SENSOR
A79C	BD B9 A0		JSR	DELAY1
A79F	BD BC 0D		JSR	RDDEFT
A7A2	7F 18 2A		CLR	COR9
A7A5	7F 18 2B		CLR	COR10
A7A8	BD BD D6		JSR	CORDIC
A7AB	FE 18 2A		LDX	COR9
A7AE	FF 18 12		STX	PHASE
A7B1	7C 18 3C		INC	SENSOR
A7B4	7C 18 3C		INC	SENSOR
A7B7	CE 1C D6		LDX	#PS1MS-10D
A7BA	FF 18 3F		STX	BCDPTR
A7BD	BD B9 B6		JSR	ADDRES
A7C0	BD BB F4		JSR	TUNE
A7C3	BD B9 A0		JSR	DELAY1
A7C6	BD BC 0D		JSR	RDDEFT
A7C9	7F 18 2A		CLR	COR9
A7CC	7F 18 2B		CLR	COR10
A7CF	BD BD D6		JSR	CORDIC
A7D2	FE 18 12		LDX	PHASE
A7D5	FF 18 00		STX	TEMP1
A7D8	FE 18 2A		LDX	COR9
A7DB	FF 18 02		STX	TEMP3
A7DE	CE 18 00		LDX	#TEMP1
A7E1	BD A7 4C		JSR	PHASDF
A7E4	EE 00		LDX	0,X

PS(REF)-PS(ALIGN)

MORE MEASURE SUBROUTINES

A7E6	FF 18 12	STX	PHASE
A7E9	39	PSS	RTS
A7EA			END

STATEMENTS =346

FREE BYTES =4319

NO ERRORS DETECTED

SUBROUTINE TO MOVE STEPPER MOTORS

		**	SUBROUTINE TO MOVE STEPPER MOTORS	
		*	USES ALL REGISTERS	
EF02		*	HARDWARE DEFINITION	
	TABLE	EQU	\$EF02	
AF88		*	EXTERNAL REFERENCE	
	GRAPH2	EQU	\$AF88	
		*	RAM DECLARATIONS	
186E	XTRAN	EQU	\$186E	
1870	YTRAN	EQU	\$1870	
1890	ANGLE	EQU	\$1890	
ED1E		ORG	\$ED1E	
ED1E	XCOUNT	RMB	5	
ED23	YCOUN	RMB	5	
ED28	RCOUNT	RMB	5	
ED2D	CTR PTR	RMB	2	
ED2F	DISPTR	RMB	2	
ED31	STEPNR	RMB	1	
ED32	TEMP1	RMB	2	
		*	PARAMETERS	
0002	XMASK	EQU	Z00000010	
0005	XCYCLE	EQU	5	
0008	YMASK	EQU	Z00001000	
0005	YCYCLE	EQU	5	
0020	RMASK	EQU	Z00100000	
0003	RCYCLE	EQU	3	
A820		ORG	\$A820	
A820	86 20	XSTEP	LDA A	#\$20
A822	B7 ED 16		STA A	INCRMT
A825	CE ED 1E		LDX	#XCOUNT
A828	FF ED 2D		STX	CTR PTR
A82B	7F ED 30		CLR	DISPTR+1
A82E	CE 18 6E		LDX	#XTRAN
A831	86 02		LDA A	#XMASK
A833	C6 05		LDA B	#XCYCLE
A835	8D 35		BSR	MV\$RST
A837	CE ED 23	YSTEP	LDX	#YCOUN
A83A	FF ED 2D		STX	CTR PTR
A83D	86 05		LDA A	5
A83F	B7 ED 30		STA A	DISPTR+1
A842	CE 18 70		LDX	#YTRAN
A845	86 08		LDA A	#YMASK
A847	C6 05		LDA B	#YCYCLE
A849	8D 21		BSR	MV\$RST
A84B	86 33	ROTATE	LDA A	#\$33
A84D	B7 ED 16		STA A	INCRMT
A850	CE ED 28		LDX	#RCOUNT
A853	FF ED 2D		STX	CTR PTR
A856	86 0A		LDA A	10
A858	B7 ED 30		STA A	DISPTR+1
A85B	CE 18 90		LDX	#ANGLE
A85E	86 20		LDA A	#RMASK
A860	C6 03		LDA B	#RCYCLE

SUBROUTINE TO MOVE STEPPER MOTORS

A862	8D 08	BSR	MU\$RST		
A864	7D ED 15	TST	CALIBR		
A867	27 74	BEQ	RETRN3		
A869	7E AF 88	JMP	GRAPH2		
A86C	37	MU\$RST	PSH B		
A86D	F6 ED 17	LDA B	PROGNR		
A870	C1 0D	CMP B	#\$D	DISPLAY ONLY?	
A872	33	PUL B			
A873	26 0C	BNE	MOVE		
A875	44	LSR A		MAKE IT CLOCK MASK	
A876	36	PSH A			
A877	FE ED 2D	LDX	CTR PTR		
A87A	A6 00	LDA A	0,X		
A87C	E6 01	LDA B	1,X		
A87E	7E AB 3D	JMP	SNMG10	TO RESTORE DISPLAY	
A881	F7 ED 31	MOVE	STA B	STEPNR	
A884	36	PSH A			
A885	E6 01	LDA B	1,X	GET NUMBER OF STEPS	
A887	A6 00	LDA A	0,X		
A889	8D 53	BSR	SINMAG		
A88B	37	PSH B		PUT BACK IN X	
A88C	36	PSH A			
A88D	30	TSX			
A88E	EE 00	LDX	0,X		
A890	31	INS			
A891	31	INS			
A892	32	PUL A			
A893	27 48	BEQ	RETRN3	NO MOTION REQUIRED	
A895	16	TAB		DIRECTION MASK	
A896	25 06	BCS	REVERS		
A898	53	FORWD	COM B		
A899	F4 EF 02	AND B	TABLE	MAKE IT ZERO	
A89C	20 03	BRA	DIRECT		
A89E	FA EF 02	REVERS	ORA B	MAKE IT 1	
A8A1	F7 EF 02	DIRECT	STA B	TABLE	
A8A4	44		LSR A	CHANGE MASK TO CLOCK	
A8A5	F6 ED 31	CYCLE	LDA B	STEPNR	
A8A8	FF ED 32		STX	TEMP1	
A8AB	37	CYCLE1	PSH B		
A8AC	CE EF 02		LDX	#\$TABLE	
A8AF	16		TAB	CLOCK MASK TO (B)	
A8B0	53		COM B		
A8B1	E4 00		AND B	0,X	
A8B3	E7 00		STA B	0,X	PULSE MOTOR
A8B5	16		TAB		
A8B6	E4 01		AND B	1,X	CONFIRM PULSE
A8B8	36		PSH A		
A8B9	AA 00		ORA A	0,X	
A8BB	A7 00		STA A	0,X	REMOVE PULSE
A8BD	32		PUL A		
A8BE	5D		TST B		
A8BF	26 28		BNE	LIMIT	NO CONFIRMATION
A8C1	C6 F0		LDA B	#\$F0	
A8C3	7D ED 15		TST	CALIBR	
A8C6	27 08		BEQ	SPEED	
A8C8	FE ED 2D		LDX	CTR PTR	
A8CB	BD AA DE		JSR	POSDIS	TO UPDATE DISPLAY

SUBROUTINE TO MOVE STEPPER MOTORS

A8CE	C6	CO	LDA B	#\$CO	EQUALIZE DELAY SET STEPPING RATE	
A8D0	BD	AA 75	JSR	DELAY1		
A8D3	33		PUL B			
A8D4	5A		DEC B			
A8D5	26	D4	BNE	CYCLE1		
A8D7	FE	ED 32	LDX	TEMP1		
A8DA	09		DEX			
A8DB	26	C8	BNE	CYCLE		
A8DD	39		RETRN3	RTS		
A8DE	0C		SINMAG	CLC		
A8DF	2A	07		BPL	PLUS	
A8E1	43			COM A		
A8E2	53			COM B		
A8E3	CB	01		ADD B	\$1	FOR 2'S COMP.
A8E5	89	00		ADC A	\$0	
A8E7	0D			SEC		
A8E8	39		PLUS	RTS		
A8E9	8E	EC FF	LIMT	LDS	#USRSTK	
A8EC	7E	AA 3A		JMP	ERROR2	
** CALLING ROUTINE FOR INITIAL ALIGNMENT						
B470		INALGN	EQU	\$B470		
A8EF	CE	A9 9B	ALIGN	LDX	#SUPRVR	
A8F2	FF	ED 1C		STX	A&TVEC	
A8F5	7F	ED 15		CLR	CALIBR	
A8F8	BD	B4 70		JSR	INALGN	
A8FB	7C	ED 15		INC	CALIBR	SO IT'S NON-ZERO
A8FE	20	10		BRA	CLEAR	

INITIALIZATION AND SUPERVISOR ROUTINES

** SETS UP PIA'S FOR SYNTHESIZERS & FLOTTER
 * FOLLOWS HARDWARE RESET

* HARDWARE DEFINITIONS

EE00	PIA	EQU	\$EE00
EE40	CONSOLE	EQU	\$EE40
EE40	KEYFAD	EQU	\$EE40

* MONITOR REFERENCES

ECFF	USRSTK	EQU	\$ECFF
EDF7	UIRQ	EQU	\$EDF7

* OTHER REFERENCES

A000	CORMAG	EQU	\$A000
A007	CORCOM	EQU	\$A007
AC47	KSPLIT	EQU	\$AC47
B47E	DOUBLE	EQU	\$B47E
B487	SINGLE	EQU	\$B487
B88C	INIZE	EQU	\$B88C

* RAM DECLARATIONS

ED00		ORG	\$ED00
ED00	LEDBFR	RMB	15
ED0F	BUFEND	EQU	*
ED0F	BUFFNT	RMB	2
ED11	KEYVAL	RMB	2
ED13	LEDPTR	RMB	2
ED15	CALIBR	RMB	1
ED16	INCRM	RMB	1
ED17	PROGMR	RMB	1
ED18	PRGJMP	RMB	2
ED1A	TEMP	RMB	2
ED1C	ABTVEC	RMB	2
ED48	GAIN	EQU	\$ED48
ED49	A\$GAIN	EQU	\$ED49
ED4A	R\$GAIN	EQU	\$ED4A
ED4B	SETDEL	EQU	\$ED4B

* PROGRAM TABLE

A800		ORG	\$A800
------	--	-----	--------

			ENTRY ADDRESS	PROGRAM NUMBER AND FUNCTION
A800	AC 47	PRGTBL	FDB	0, X-Y PLOT FROM TEST IMAGE
A802	AC 47		FDB	1, CRT DISPLAY FROM TEST IMAGE
A804	AC 47		FDB	2, X-Y PLOT FROM REF. IMAGE
A806	AC 47		FDB	3, CRT DISPLAY FROM REF. IMAGE
A808	A8 EF		FDB	4, INITIAL ALIGNMENT, 2 SENS
A80A	A8 EF		FDB	5, INITIAL ALIGNMENT, 1 SENS
A80C	B4 7E		FDB	6, TEST ALIGNMENT, 2 SENSORE
A80E	B4 7E		FDB	7, TEST ALIGNMENT, 2 SENS.,
A810	B4 87		FDB	8, TEST ALIGNMENT, 1 SENSOR

INITIALIZATION AND SUPERVISOR ROUTINES

A812	B4	B7	FDB	SINGLE	9, TEST ALIGNMENT, 1 SENSOR,
A814			RMB	2	RESERVED FOR ABORT
A816	A0	00	FDB	CORMAG	B, MAGNITUDE CORRELATION
A818	A0	07	FDB	CORCOM	C, COMPLEX CORRELATION
A81A	A8	20	FDB	XSTEP	D, DISPLAY TABLE POSITION
A81C			RMB	2	RESERVED FOR EXECUTE
A81E	A9	6A	FDB	INITLZ	F, RE-INITIALIZE PARAMETERS

* SEGMENT TABLE

A900			ORG	\$A900
A900	B1	CF	FCB	\$81,\$CF,\$92,\$86,\$CC,\$A4,\$A0,\$8F
A908	80	8C	FCB	\$80,\$8C,\$88,\$E0,\$B1,\$C2,\$B0,\$B8

** SETS POSITION COUNTERS TO ZERO

A910	CE	ED	1E	CLEAR	LDX	\$XCOUNT
A913	6F	00		CLEAR1	CLR	0,X
A915	08				INX	
A916	8C	ED	2D		CPX	\$XCOUNT+15
A919	26	F8			BNE	CLEAR1
A91B	39				RTS	

** HARDWARE RESET VECTORS HERE

A91C	01		ARESET	NOF		
A91D	CE	EE	07	LDX	\$PIA+7	
A920	EF	00		STX	0,X	
A922	09		ARSET1	DEX		
A923	26	FD		BNE	ARSET1	
A925	CE	EE	00	LDX	\$PIA	
A928	35			TXS		
A929	86	41		LDA A	\$\$41	7+E+2,1/16,RTS FALSE
A92B	A7	08		STA A	8,X	
A92D	86	FF		LDA A	\$\$FF	
A92F	A7	10		STA A	\$10,X	ALL OUTPUTS
A931	A7	11		STA A	\$11,X	
A933	A7	20		STA A	\$20,X	
A935	A7	21		STA A	\$21,X	
A937	A7	41		STA A	\$41,X	
A939	A7	80		STA A	\$80,X	
A93B	A7	81		STA A	\$81,X	
A93D	86	36		LDA A	#\$00110110	TO SENSE POSITIVE BIAS TRANS
A93F	A7	12		STA A	\$12,X	THRU CA1
A941	A7	13		STA A	\$13,X	LINE TRANSITION THRU CR1
A943	86	04		LDA A	#\$00000100	TO SENSE NEG. TRANSITION
A945	A7	22		STA A	\$22,X	
A947	A7	82		STA A	\$82,X	
A949	86	34		LDA A	#\$00110100	CB2 AS LOW OUTPUT
A94B	A7	23		STA A	\$23,X	
A94D	A7	83		STA A	\$83,X	
A94F	A7	42		STA A	\$42,X	CA2 HERE
A951	86	2D		LDA A	#\$00101101	
A953	A7	43		STA A	\$43,X	PULSE MODE, INTERRUPTS ON
A955	CE	EF	00	LDX	\$PIA+\$100	
A958	86	CF		LDA A	#\$FF-\$30	
A95A	A7	00		STA A	0,X	TURN AROUND LS245'S

INITIALIZATION AND SUPERVISOR ROUTINES

A95C	6F 02	CLR	2,X	AND THEN PIA DDR	
A95E	86 CB	LDA A	#\$FF-\$34		
A960	A7 00	STA A	0,X		
A962	86 C3	LDA A	#\$FF-\$3C		
A964	A7 01	STA A	1,X	A-SIDE REMAINS INPUTS	
A966	86 FF	LDA A	#\$FF	B-SIDE OUTPUTS LOW	
A968	A7 02	STA A	2,X		
A96A	86 ED	INITLZ	LDA A	#LEDBFR/256	
A96C	87 ED 2F		STA A	DISPTR	
A96F	86 A9		LDA A	#LEDTBL/256	
A971	B7 ED 13		STA A	LEIPTR	
A974	86 7E		LDA A	#\$7E	JMP INSTRUCTION
A976	B7 ED F7		STA A	UIREQ	
A979	CE AA A5		LDX	#INTSRV	SET INTERRUPT VECTOR
A97C	FF ED F8		STX	UIREQ+1	
A97F	8D 8F		BSR	CLEAR	
A981	86 A8		LDA A	#FRGTBL/256	
A983	B7 ED 18		STA A	PRGJMP	
A986	7F ED 15		CLR	CALIBR	
A989	CE 00 01		LDX	#1	
A98C	FF ED 48		STX	SETDEL	
A98F	86 03		LDA A	#3	
A991	B7 ED 49		STA A	A\$GAIN	
A994	4A		DEC A		
A995	B7 ED 4A		STA A	R\$GAIN	
A998	BD B8 8C		JSR	INIZE	
A99B	CE ED 00	SUPRVR	LDX	#LEDBFR	
A99E	FF ED 0F		STX	BUFFNT	
A9A1	86 FE		LDA A	#\$FE	
A9A3	A7 00	DASH	STA A	0,X	DISPLAY DASHES
A9A5	08		INX		
A9A6	8C ED 0F		CPX	#BUFEND	
A9A9	26 F8		BNE	DASH	
A9AB	86 FF		LDA A	#\$FF	
A9AD	B7 ED 17		STA A	PROGNR	
A9B0	8E EC FF		LDS	#USRSTK	
A9B3	CE A9 98		LDX	#SUPRVR	
A9B6	FF ED 1C		STX	ABTVEC	

* WAIT FOR KEYPAD INPUT & PROCESS COMMAND

A9B9	01	WAITLP	NOP		
A9BA	0E		CLI		
A9BB	01		NOP		
A9BC	86 EE 40		LDA A	KEYPAD	
A9BF	43		COM A		
A9C0	27 45		REQ	MANDSP	NO KEY PRESSED
A9C2	CE ED 11		LDX	#KEYVAL	
A9C5	BD AA 7B		JSR	KEYIN	
A9C8	E6 00		LDA B	0,X	
A9CA	2B 3B		BMI	MANDSP	INVALID KEY VALUE
A9CC	C1 0A		CMP B	#\$A	ABORT KEY?
A9CE	27 37		REQ	MANDSP	
A9D0	C1 0E		CMP B	#\$E	EXECUTE?
A9D2	26 19		BNE	PRGDSP	
A9D4	86 ED 17		LDA A	PROGNR	
A9D7	2B E0		BMI	WAITLP	NO PROGRAM NUMBER

INITIALIZATION AND SUPERVISOR ROUTINES

A9D9	48	ASL A		DOUBLE IT	
A9DA	B7 ED 19	STA A	PRGJMP+1		
A9DD	FE ED 18	LDX	PRGJMP		
A9E0	EE 00	LDX	0,X	PROGRAM VECTOR	
A9E2	AD 00	JSR	0,X	TO EXECUTE	
A9E4	B6 ED 17	LDA A	PROG NR		
A9E7	81 0B	CMP A	#\$B		
A9E9	24 CE	BCC	WAITLP		
A9EB	20 AE	BRA	SUPRVR		
A9ED	F7 ED 14	PRGDSP	STA B	DISPLAY PROGRAM NUMBER	
A9F0	FE ED 13	LDX	LEDPTR		
A9F3	A6 00	LDA A	0,X		
A9F5	CE ED 00	LDX	#\$LEDBFR		
A9F8	A7 00	STA A	0,X		
A9FA	F7 ED 17	STA B	PROG NR		
A9FD	86 FF	LDA A	#\$FF		
A9FF	08	CLEAR2	INX		
AA00	A7 00	STA A	0,X		
AA02	8C ED 0E	CPX	#\$BUFEND-1		
AA05	26 F8	BNE	CLEAR2		
AA07	7D ED 15	MANDSP	TST	CALIBR	INITIAL ALIGNMENT DONE?
AA0A	27 AD	BEQ	WAITLP		
AA0C	86 15	LDA A	#\$X00010101	MASK FOR ANY CLOCK	
AA0E	B4 EF 03	AND A	TABLE+1		
AA11	88 15	EOR A	#\$X00010101		
AA13	27 A4	BEQ	WAITLP	NO PULSE	
AA15	C6 05	LDA B	#\$5		
AA17	8D 5E	BSR	DELAY2		
AA19	16	TAB			
AA1A	F4 EF 03	AND B	TABLE+1	CONFIRM IT	
AA1D	26 9A	BNE	WAITLP		
AA1F	36	PSH A			
AA20	C6 03	LDA B	#\$3		
AA22	CE ED 1E	LDX	#\$XCOUNT		
AA25	44	IDENT	LSR A		FIND OUT WHICH AXIS
AA26	25 28		BCS		
AA28	44		LSR A		
AA29	08		INCSET		
AA2A	08				
AA2B	08				
AA2C	08				
AA2D	08				
AA2E	5A				
AA2F	26 F4				
AA31	C6 02	ERROR	BNE	IDENT	
AA33	F7 ED 30		LDA B	#\$2	
AA36	20 02		STA B	DISPTR+1	
AA38	8D 24	ERROR1	BRA	ERROR2	
AA3A	FE ED 2F	ERROR2	BSR	SETUP	
AA3D	C6 B0		LDX	DISPTR	
AA3F	E7 00		LDA B	#\$B0	"E"
AA41	C6 FA		STA B	0,X	
AA43	E7 01		LDA B	#\$FA	LOWER CASE "R"
AA45	E7 02		STA B	1,X	
AA47	E7 04		STA B	2,X	
AA49	C6 E2		STA B	4,X	
AA4B	E7 03		LDA B	#\$E2	"O"
			STA B	3,X	

INITIALIZATION AND SUPERVISOR ROUTINES

AA4D	7E A9 B9	WAITEX	JMP	WAITLP	
AA50	8D 0C	INCSET	BSR	SETUP	
AA52	32		FUL A		RETRIEVE CLOCK MASK
AA53	BD AA DE		JSR	POSDIS	
AA56	16	CLKLOW	TAB		
AA57	F4 EF 03		AND B	TABLE+1	WAIT FOR END OF PULSE
AA5A	27 FA		REQ	CLKLOW	
AA5C	20 EF		BRA	WAITEX	
AA5E	50	SETUP	NEG B		
AA5F	CB 03		ADD B	\$3	
AA61	86 20		LDA A	\$\$20	5 STEPS/COUNT FOR X OR Y
AA63	C1 02		CMP B	\$2	
AA65	2B 02		BMI	STEPST	
AA67	86 33		LDA A	\$\$33	3 STEPS/COUNT FOR R
AA69	B7 ED 16	STEPST	STA A	INCRMT	
AA6C	17		TBA		MULTIPLY BY 5
AA6D	48		ASL A		
AA6E	48		ASL A		
AA6F	1B		ABA		
AA70	B7 ED 30		STA A	DISPTR+1	LED BUFFER OFFSET
AA73	39		RTS		
** DELAY SUBROUTINES					
AA74	5F	DELAY0	CLR B		
AA75	8D 00	DELAY1	BSR	DELAY2	
AA77	5A	DELAY2	DEC B		
AA78	26 FD		BNE	DELAY2	
AA7A	39		RTS		
** DEBOUNCE/DECODE ROUTINE					
AA7B	86 FF	KEYIN	LDA A	\$\$FF	
AA7D	A7 00		STA A	0,X	DEFAULT VALUE
AA7F	B8 EE 40		EOR A	KEYPAD	TO INVERT
AA82	27 20		BEQ	RETURN	NO KEY PRESSED
AA84	8D EE		BSR	DELAY0	FOR DEBOUNCE
AA86	B6 EE 40		LDA A	KEYPAD	
AA89	8D E9		BSR	DELAY0	
AA8B	16		TAB		
AA8C	F8 EE 40		EOR B	KEYPAD	SAME KEY?
AA8F	26 13		BNE	RETURN	READ ERROR
AA91	36		FSH A		
AA92	C6 04		LDA B	\$4	
AA94	5A	COLUMN	DEC B		
AA95	48		ASL A		
AA96	25 FC		BCS	COLUMN	
AA98	32		FUL A		
AA99	47	ROW	ASR A		
AA9A	24 06		BCC	DONE	
AA9C	CB 04		ADD B	\$4	
AA9E	25 04		BCS	RETURN	DECODE ERROR
AAA0	20 F7		BRA	ROW	
AAA2	E7 00	DONE	STA B	0,X	
AAA4	39	RETURN	RTS		

INITIALIZATION AND SUPERVISOR ROUTINES

** INTERRUPT SERVICE ROUTINE

AAA5	FE ED 0F	INTSRV	LDX	BUFFNT	
AAA8	8C ED 0F		CPX	#BUFEND	
AAA8	26 0D		BNE	CHAN1	NOT FAST LAST DIGIT
AAAD	CE ED 00		LDX	#LEDBFR	
AAE0	86 3C		LDA A	\$Z00111100	RESET DISPLAY COUNT
AAB2	B7 EE 42		STA A	CONSOLE+2	
AAB5	86 34		LDA A	\$Z00110100	
AAB7	B7 EE 42		STA A	CONSOLE+2	
AABA	A6 00	CHAN1	LDA A	0,X	
AABC	B7 EE 41		STA A	CONSOLE+1	OUTPUT NEW DIGIT
AABF	08		INX		
AAC0	FF ED 0F		STX	BUFFNT	STORE NEW POINTER
AAC3	7D EE 41		TST	CONSOLE+1	CLEAR INTERRUPT REQUEST
AAC6	B6 EE 40		LDA A	KEYPAD	
AAC9	81 BB		CMP A	#\$BB	IS "A" PRESSED?
AACB	26 10		BNE	RETRN1	NO
AACD	CE ED 12		LDX	#KEYVAL+1	
AAD0	8D A9		BSR	KEYIN	BE SURE IT'S "A"
AAD2	A6 00		LDA A	0,X	
AAD4	81 0A		CMP A	#\$A	
AAD6	26 05		BNE	RETRN1	FALSE ALARM
AAD8	FE ED 1C		LDX	ABTVEC	GET ABORT VECTOR
AADB	6E 00		JMP	0,X	
AADD	3B	RETRN1	RTI		
AADE		POSDIS	EQU	*	
AB3D		SNMG10	EQU	*+\$5F	
AADE			END		

STATEMENTS =466

FREE BYTES =1263

NO ERRORS DETECTED

-- ::

+1,\$ASSEMBLE

STEPPER MOTOR POSITION DISPLAY
FOR IMAGE ALIGNMENT SYSTEM

* EXTERNAL REFERENCES

EF02	TABLE	EQU	\$EF02	
ED13	LEDPTR	EQU	\$ED13	
ED16	INCRMT	EQU	\$ED16	
ED1A	TEMP	EQU	\$ED1A	
ED2F	DISPTR	EQU	\$ED2F	
AADE		ORG	\$AADE	
AADE 36	POSDIS	PSH A		STORE CLOCK MASK
AAIF 48		ASL A		
AAE0 16		TAB		
AAE1 B4 EF 03		AND A	TABLE+1	WHICH DIRECTION?
AAE4 C5 08		BIT B	#Z00001000	
AAE6 26 06		BNE	Y	
AAE8 F7 ED 1A		STA B	TEMP	
AAE9 B8 ED 1A		EOR A	TEMP	SO DISPLAY AGREES WITH DIREC
AAEE 4D	Y	TST A		
AAEF 27 0C		BEQ	INCR1	
AAF1 86 FF	DECR	LDA A	#\$FF	REVERSE
AAF3 36		PSH A		
AAF4 36		PSH A		
AAF5 86 99		LDA A	#\$99	FOR BCD UPDATE
AAF7 B0 ED 16		SUB A	INCRMT	
AAFA 0D		SEC		
AAFB 20 09		BRA	ADD1	
AAFD 4F	INCR1	CLR A		FORWARD
AAFE 36		PSH A		
AAFF 86 01		LDA A	#1	
AB01 36		PSH A		
AB02 B6 ED 16		LDA A	INCRMT	
AB05 5F		CLR B		
AB06 A9 02	ADD1	ADC A	2,X	BCD UPDATE
AB08 19		DAA		
AB09 25 0A		BCS	TEST1	TAKE CARE OF ROUNDOFF ERROR
AB0B 81 99		CMP A	#\$99	
AB0D 0C		CLC		
AB0E 26 0B		BNE	STORE	
AB10 8B 01		ADD A	#1	ROUND UP TO 100
AB12 19		DAA		
AB13 20 06		BRA	STORE	
AB15 81 01	TEST1	CMP A	#1	
AB17 26 01		BNE	SETCRY	
AB19 4F		CLR A		ROUND DOWN TO 0
AB1A 0D	SETCRY	SEC		
AB1B A7 02	STORE	STA A	2,X	
AB1D 86 00		LDA A	#0	ASSUME FORWARD
AB1F C4 FF		AND B	#\$FF	
AB21 27 02		BEQ	ADD2	
AB23 86 99		LDA A	#\$99	NO, REVERSE
AB25 16	ADD2	TAB		
AB26 A9 01		ADC A	1,X	
AB28 19		DAA		
AB29 A7 01		STA A	1,X	
AB2B 17		TBA		
AB2C A9 00		ADC A	0,X	
AB2E 19		DAA		

STEPPER MOTOR POSITION DISPLAY
FOR IMAGE ALIGNMENT SYSTEM

AB2F	A7 00	STA A	0,X		
AB31	33	PUL B		UPDATE BINARY COUNT	
AB32	EB 04	ADD B	4,X		
AB34	E7 04	STA B	4,X		
AB36	33	PUL B			
AB37	E9 03	ADC B	3,X		
AB39	E7 03	STA B	3,X		
AB3B	E6 01	LDA B	1,X	BCD COUNT NOW IN (AB)	
AB3D	FF ED 1A	SNMG10	STX TEMP	CONVERT TO SIGN/MAGNITUDE	
AB40	EE 00	LDX 0,X			
AB42	8C 50 00	CPX #\$5000			
AB45	0C	CLC			
AB46	2B 1E	BMI FIXSGN		IT'S POSITIVE	
AB48	FE ED 1A	MINUS1	LDX TEMP	IT'S NEGATIVE	
AB4B	86 99	LDA A	#\$99		
AB4D	16	TAB			
AB4E	E0 01	SUB B	1,X		
AB50	F7 ED 1B	STA B	TEMP+1		
AB53	A0 00	SUB A	0,X		
AB55	B7 ED 1A	STA A	TEMP		
AB58	4F	CLR A			
AB59	0D	SEC			
AB5A	B9 ED 1B	ADC A	TEMP+1		
AB5D	19	DAA			
AB5E	16	TAB			
AB5F	86 00	LDA A	\$0		
AB61	B9 ED 1A	ADC A	TEMP		
AB64	19	DAA			
AB65	0D	SEC		TO MARK SIGN	
AB66	36	FIXSGN	PSH A	FIX SIGN ON DISPLAY	
AB67	86 00	LDA A	\$0		
AB69	49	ROL A		GET CARRY BIT	
AB6A	43	COM A		CARRY SET MAKES MINUS	
AB6B	FE ED 2F	LDX DISPTR		LEFTMOST DIGIT	
AB6E	A7 00	STA A	0,X		
AB70	32	PUL A		M. S. DIGITS	
AB71	37	PSH B			
AB72	8D 20	BSR LOOKUP			
AB74	FE ED 2F	LDX DISPTR			
AB77	08	INX			
AB78	8D 32	BSR DISPLA			
AB7A	32	PUL A		L. S. DIGITS	
AB7B	8D 17	BSR LOOKUP			
AB7D	FE ED 2F	LDX DISPTR			
AB80	08	INX			
AB81	08	INX			
AB82	08	INX			
AB83	8D 27	BSR DISPLA			
AB85	32	PUL A			
AB86	81 10	CMP A	#\$200010000	IS IT ANGLE?	
AB88	27 03	REQ ANGLPT			
AB8A	22 07	BHI RETRN2		IT'S CORRELATION COEFFICIENT	
AB8C	09	DEX		POSITION D. P.	
AB8D	E6 00	ANGLPT	LDA B	0,X	
AB8F	C4 7F		AND B	#\$7F	TO ADD D. P.
AB91	E7 00		STA B	0,X	
AB93	39	RETRN2	RTS		

STEPPER MOTOR POSITION DISPLAY
FOR IMAGE ALIGNMENT SYSTEM

AB94	16	LOOKUP	TAB	
AB95	44		LSR A	
AB96	44		LSR A	
AB97	44		LSR A	
AB98	44		LSR A	
AB99	B7 ED 14		STA A	LEDPTR+1 FIRST DIGIT
AB9C	FE ED 13		LDX	LEDPTR
AB9F	A6 00		LDA A	0,X GET SEGMENT CODE
ABA1	C4 0F		AND B	#\$0F SECOND DIGIT
ABA3	F7 ED 14		STA B	LEDPTR+1
ABA6	FE ED 13		LDX	LEDPTR
ABA9	E6 00		LDA B	0,X
ABAB	39		RTS	
ABAC	48	DISPLA	ASL A	
ABAD	68 00		ASL	0,X SAVE D. P. IN 'C'
ABAF	A7 00		STA A	0,X
ABB1	66 00		ROR	0,X RESTORE D. P.
ABB3	58		ASL B	
ABB4	68 01		ASL	1,X
ABB6	E7 01		STA B	1,X
ABB8	66 01		ROR	1,X
ABBA	39		RTS	
ABBB			END	

STATEMENTS =142

FREE BYTES =7297

NO ERRORS DETECTED

--
--

^1,\$ASSEMBLE

X-Y PLOTTER/CRT DISPLAY PROGRAM
FOR IMAGE ALIGNMENT SYSTEM

** SUPERVISOR REFERENCES

ED1A	TEMP	EQU	\$ED1A
ED17	PROGNR	EQU	\$ED17
ED1C	ABTVEC	EQU	\$ED1C
A99B	SUPRVR	EQU	\$A99B
BDD6	CORDIC	EQU	\$BDD6

** RAM DECLARATIONS

1700		ORG	\$1700
1700	VMAX	RMB	256
1824	COR1	EQU	\$1824
ED34		ORG	\$ED34
ED34	VMXFTR	RMB	1
ED35	XCOORD	RMB	2
ED37	YCOORD	RMB	2
ED39	SLOW	RMB	2
ED3B	XNMBR	RMB	1
ED3C	YNMBR	RMB	1
ED3D	YPOSN	RMB	1
ED3E	XMSG	RMB	4
ED42	CTR	RMB	1
ED43	FREQ	RMB	4
ED47	SIGN	RMB	1
ED48	GAIN	RMB	1
ED49	A\$GAIN	RMB	1
ED4A	R\$GAIN	RMB	1
ED4B	SETDEL	RMB	2

** HARDWARE DEFINITIONS

DFF0	DAS	EQU	\$DFF0
EE08	ACIASR	EQU	\$EE08
EE00	PIA	EQU	\$EE00
EE80	VERT	EQU	PIA+\$80
EE81	HORIZ	EQU	VERT+1
EE10	XFREQ	EQU	PIA+\$10
EE20	YFREQ	EQU	PIA+\$20

** FANTOM-II REFERENCES

FD7A	OUT4HS	EQU	\$FD7A
FD7E	OUTSP	EQU	\$FD7E
FD80	OUTCH	EQU	\$FD80
FDFF	OUTSTR	EQU	\$FDFF
FDA9	CRLF1	EQU	\$FDA9
FD8F	THR	EQU	\$FD8F
FD9C	THR1	EQU	\$FD9C

** ASCII MESSAGE DATA

ABCO		ORG	\$ABCO	
ABCO	20 53 54	STMSG	FCC	' STARTING FREQ' ' ' ' ' '\$+\$80
ABDO	20 4D 48	MHZMSG	FCC	' MHz' ' ' '\$80

X-Y PLOTTER/CRT DISPLAY PROGRAM
FOR IMAGE ALIGNMENT SYSTEM

ABD5	0D 0A 1B	ESC\$FF	FCC	\$0D,\$0A,\$1B,\$0C+\$80	
ABD9	54 45 53	TSTMSG	FCC	'TEST IMAG','E'+\$80	
ABE3	52 45 46	REFMSG	FCC	'REFERENCE IMAG','E'+\$80	
ABF2	1F 87	SIGNAL	FCC	\$1F,7+\$80	
ABF4	CE AB D5	PAGE	LDX	#ESC\$FF	SEND CR, LF, AND ESC/FF
ABF7	BD FD FF		JSR	OUTSTR	
ABFA	CE 00 00		LDX	#0	
ABFD	BD AD 7A		JSR	DELAY	1 SECOND TO ERASE CRT
AC00	7E AD 7A		JMP	DELAY	RETURNS VIA RTS IN DELAY
AC03	8D EF	OUTFRQ	BSR	PAGE	
AC05	CE AB C0		LDX	#\$TMSG	
AC08	BD FD FF		JSR	OUTSTR	
AC0B	86 3A		LDA A	#';'	
AC0D	BD FD 80		JSR	OUTCH	
AC10	CE 20 46		LDX	#' F'	
AC13	FF ED 3E		STX	XMSG	
AC16	CE 78 BD		LDX	#'x='+\$80	
AC19	FF ED 40		STX	XMSG+2	
AC1C	B6 ED 46		LDA A	FREQ+3	
AC1F	8D 1A		BSR	SHUFL	
AC21	08		INX		
AC22	08		INX		
AC23	8D 08		BSR	OUTFR1	
AC25	7C ED 40		INC	XMSG+2	CHANGE TO Y
AC28	B6 ED 44		LDA A	FREQ+1	
AC2B	8D 0E		BSR	SHUFL	
AC2D	86 2E	OUTFR1	LDA A	#'.'	
AC2F	BD FD 80		JSR	OUTCH	
AC32	BD FD 8F		JSR	THB	
AC35	CE AB D0		LDX	#MHZMSG	
AC38	7E FD FF		JMP	OUTSTR	
AC3B	B7 ED 42	SHUFL	STA A	XMSG+4	(USE 'CTR' TEMPORARILY)
AC3E	CE ED 3E		LDX	#XMSG	
AC41	BD FD FF		JSR	OUTSTR	
AC44	7E FD 8F		JMP	THB	

** MAIN PROGRAM

AC47	CE AD 01	NSPLOT	LDX	#PABORT	LOAD ABORT VECTOR
AC4A	FF ED 1C		STX	ABTVEC	
AC4D	CE 58 29		LDX	#\$5829	LOAD STARTING FREQ'S.
AC50	FF ED 43		STX	FREQ	
AC53	CE 26 32		LDX	\$2632	
AC56	FF ED 45		STX	FREQ+2	
AC59	FE ED 43		LDX	FREQ	
AC5C	FF EE 20		STX	YFREQ	
AC5F	FE ED 45		LDX	FREQ+2	
AC62	FF EE 10		STX	XFREQ	
AC65	8D 9C		BSR	OUTFRQ	
AC67	CE 17 00		LDX	#VMAX	
AC6A	FF ED 34		STX	VMXPTR	

X-Y PLOTTER/CRT DISPLAY PROGRAM
FOR IMAGE ALIGNMENT SYSTEM

AC6D	6F 00	ZERO	CLR	0,X	
AC6F	08		INX		
AC70	8C 18 00		CPX	\$UMAX+256	
AC73	26 F8		BNE	ZERO	
AC75	CE ED 35		LDX	*XCOORD	
AC78	6F 00	ZERO1	CLR	0,X	
AC7A	08		INX		
AC7B	8C ED 3E		CPX	*XMSG	
AC7E	26 F8		BNE	ZERO1	
AC80	7F EE 80		CLR	VERT	
AC83	CE AB D9		LDX	*TSTMMSG	
AC86	B6 ED 49		LDA A	A\$GAIN	
AC89	B7 ED 48		STA A	GAIN	
AC8C	B6 ED 17		LDA A	PROGMR	
AC8F	84 02		AND A	*\$00000010	
AC91	B7 DF F9		STA A	DAS+9	SELECT DAS INPUT
AC94	27 09		BEQ	IDENT	
AC96	CE AB E3		LDX	*REFMSG	
AC99	B6 ED 4A		LDA A	R\$GAIN	
AC9C	B7 ED 48		STA A	GAIN	
AC9F	C6 0C	IDENT	LDA B	*\$12	
ACA1	BD AF 3A		JSR	SPSTR1	
ACA4	4F		CLR A		
ACAS	B7 ED 35	RETRCE	STA A	XCOORD	
ACAB	BD AD 60		JSR	PROGTST	
ACAB	27 6E		BEQ	HFOSN	
ACAD	BD AE 25		JSR	DARK	
ACB0	CE 20 00		LDX	*\$2000	
ACB3	BD AD 7A		JSR	DELAY	
ACB6	20 6F		BRA	CONVRT	
ACB8	7C ED 3B	XCHECK	INC	XNMBR	
ACBB	F6 ED 3B		LDA B	XNMBR	
ACRE	C1 C0		CMP B	*\$192	END OF LINE?
ACCO	26 65		BNE	CONVRT	
ACC2	FE ED 45		LDX	FREQ+2	
ACC5	FF EE 10		STX	XFREQ	
ACC8	7F ED 3B		CLR	XNMBR	
ACCB	CE EE 20		LDX	*YFREQ	
ACCE	86 12		LDA A	*\$12	
ACD0	BD 34		BSR	DECADD	
ACD2	7C ED 3C		INC	YNMBR	
ACD5	BD AD 66		JSR	LFTPEN	
ACD8	B6 ED 3D		LDA A	YPOSN	
ACDB	8B 02		ADD A	*2	
ACDD	BD AD 60		JSR	PROGTST	
ACE0	26 02		BNE	YCMFR	
ACE2	8B 01		ADD A	*1	
ACE4	B7 ED 3D	YCMFR	STA A	YPOSN	
ACE7	B7 EE 80		STA A	VERT	
ACEA	B6 ED 3C		LDA A	YNMBR	
ACED	81 40		CMP A	*\$64	
ACEF	26 B4		BNE	RETRCE	
ACF1	BD 73	DONE	BSR	LFTPEN	
ACF3	7F EE 80		CLR	VERT	
ACF6	86 FF		LDA A	*\$FF	
ACF8	B7 EE 81		STA A	HORIZ	
ACFB	CE AB F2		LDX	*\$IGNAL	

X-Y PLOTTER/CRT DISPLAY PROGRAM
FOR IMAGE ALIGNMENT SYSTEM

ACFE	7E FD FF		JMP	OUTSTR	RETURNS VIA RTS
AD01	8D EE	PABORT	BSR	DONE	
AD03	7E A9 9B		JMP	SUFRVR	
AD06	AB 00	DECADD	ADD A	0,X	
AD08	19		DAA		
AD09	A7 00		STA A	0,X	
AD0B	A6 01		LDA A	1,X	
AD0D	89 00		ADC A	#0	
AD0F	19		DAA		
AD10	A7 01		STA A	1,X	
AD12	39		RTS		
AD13	7A EE 81	PLOTRT	DEC	HORIZ	
AD16	CE 02 92		LDX	#\$292	
AD19	8D 5F		BSR	DELAY	
AD1B	B1 EE 81	HPOSN	CMP A	HORIZ	
AD1E	26 F3		BNE	PLOTRT	
AD20	CE 80 00		LDX	#\$8000	
AD23	8D 55		BSR	DELAY	
AD25	8D 4B		BSR	DROPEN	
AD27	CE 18 24	CONVRT	LDX	#\$COR1	
AD2A	BD AE 2A		JSR	SYNDET	
AD2D	CE EE 10		LDX	#\$XFREQ	
AD30	86 04		LDA A	#\$04	
AD32	E6 01		LDA B	1,X	
AD34	C0 30		SUB B	#\$30	
AD36	8D CE		BSR	DECADD	
AD38	EA 01		ORA B	1,X	
AD3A	C4 30		AND B	#\$30	CHECK X FREQ CHANGE
AD3C	26 05		BNE	DISPL1	
AD3E	CE FF FF		LDX	#\$FFFF	39 TO 40 OR 79 TO 80
AD41	8D 37		BSR	DELAY	
AD43	FE ED 4B	DISPL1	LDX	SETDEL	OPTIONAL DELAY
AD46	8D 32		BSR	DELAY	FOR SYNTH. SETTLING
AD48	CE 18 24		LDX	#\$COR1	
AD4B	6F 06		CLR	6,X	
AD4D	6F 07		CLR	7,X	
AD4F	BD BD D6		JSR	CORDIC	
AD52	F6 ED 4B		LDA B	GAIN	
AD55	78 18 25	BOOST	ASL	COR1+1	
AD58	79 18 24		ROL	COR1	
AD5B	5A		DEC B		
AD5C	26 F7		BNE	BOOST	
AD5E	20 1E		BRA	MASK	
AD60	F6 ED 17	PROGTST	LDA B	PROG NR	
AD63	C5 01		BIT B	#1	
AD65	39		RTS		
AD66	CE 80 00	LFTPEN	LDX	#\$8000	
AD69	8D 0F		BSR	DELAY	
AD6B	86 34		LDA A	#\$34	
AD6D	B7 EE 83		STA A	VERT+3	
AD70	20 05		BRA	DELAY1	
AD72	C6 3C	DROPEN	LDA B	#\$3C	
AD74	F7 EE 83		STA B	VERT+3	

LIFT PEN

X-Y PLOTTER/CRT DISPLAY PROGRAM
FOR IMAGE ALIGNMENT SYSTEM

AD77	CE 10 00	DELAY1	LDX	#\$1000	
AD7A	09	DELAY	DEX		
AD7B	26 FD		BNE	DELAY	
AD7D	39		RTS		
AD7E	B6 18 24	MASK	LDA A	COR1	
AD81	F6 18 25		LDA B	COR1+1	
AD84	F7 ED 38		STA B	YCOORD+1	
AD87	BB ED 30		ADD A	YPOSN	VERTICAL CHANGE
AD8A	24 02		BCC	NOTSAT	
AD8C	86 FF		LDA A	#\$FF	
AD8E	FE ED 34	NOTSAT	LDX	VMXPTR	VMXPTR+1=XCOORD
AD91	E6 00		LDA B	0,X	
AD93	11		CBA		
AD94	24 11		BCC	BIGR	NEW IS HIGHER
AD96	37		PSH B		
AD97	8D C7		BSR	PROGTST	
AD99	26 48		BNE	BLANK	
AD9B	86 08		LDA A	#\$00001000	CHECK PEN
AD9D	B5 EE B3		BIT A	VERT+3	
ADA0	27 02		BEQ	KEEPB	UP ALREADY
ADA2	8D C2		BSR	LFTPEN	
ADA4	32	KEEPB	PUL A	SMALR	KEEP OLD ELEV.
ADA5	20 12		BRA	PROGTST	
ADA7	8D B7	BIGR	BSR	CNTNUE	
ADA9	26 09		BNE	#\$00001000	CHECK PEN
ADAB	C6 08		LDA B	VERT+3	
ADAD	F5 EE B3		BIT B	CNTNUE	
ADBD	26 02		BNE	DROPEN	
ADB2	8D BE		BSR	VMXPTR	
ADB4	FE ED 34	CNTNUE	LDX	0,X	
ADB7	A7 00		STA A	YCOORD	
ADB9	B7 ED 37	SMALR	STA A	XCOORD	
ADBC	7C ED 35		INC	PROGTST	
ADBF	8D 9F		BSR	WRITE	
ADC1	26 1C		BNE		
ADC3	16	SLWADJ	TAB		
ADC4	F0 EE 80		SUB B	VERT	GET VERT CHANGE
ADC7	24 01		BCC	POS	
ADC9	50		NEG B		
ADCA	58	POS	ASL B		MULTIPLY BY 4
ADCB	58		ASL B		
ADCC	CB 08		ADD B	\$8	MINIMUM PLOT DELAY
ADCE	F7 ED 39		STA B	SLOW	
ADD1	FE ED 39		LDX	SLOW	
ADD4	8D A4		BSR	DELAY	
ADD6	B7 EE 80		STA A	VERT	
ADD9	7C EE B1	XCHJMP	INC	HORIZ	
ADDc	7E AC B8		JMP	XCHECK	
ADDf	8D 11	WRITE	BSR	VECTOR	
ADE1	20 F9		BRA	XCHJMP	

X-Y PLOTTER/CRT DISPLAY PROGRAM
FOR IMAGE ALIGNMENT SYSTEM

** FORMAT DATA FOR OUTPUT TO TEK 4006-1

ADE3	31	BLANK	INS		
ADE4	A6 00		LDA A	0,X	
ADE6	B7 ED 37		STA A	YCOORD	
ADE9	7C ED 35		INC	XCOORD	
ADEC	8D 02		BSR	DRKVCT	
ADEE	20 EC		BRA	XCHJMP	
ADF0	8D 33	DRKVCT	BSR	DARK	
ADF2	CE E1 75	VECTOR	LDX	\$XCOORD	
ADF5	8D 1B		BSR	LOBYTE	
ADF7	8B C0		ADD A	\$\$40+\$80	MARK END OF STRING
ADF9	36		PSH A		LAST WORD OUT
ADFA	8D 21		BSR	HIBYTE	
ADFC	36		PSH A		
ADFD	CE ED 37	LOY	LDX	\$YCOORD	
AE00	8D 10		BSR	LOBYTE	
AE02	8B 60		ADD A	#\$60	ADD TAG
AE04	36		PSH A		
AE05	8D 16		BSR	HIBYTE	
AE07	4C		INC A		MOVE RASTER UP 32 LINES
AE08	36		PSH A		FIRST WORD OUT
AE09	30		TSX		
AE0A	8D FD FF		JSR	OUTSTR	SEND VECTORS TO DISPLAY
AE0D	31		INS		RESTORE S.P.
AE0E	31		INS		
AE0F	31		INS		
AE10	31		INS		
AE11	39		RTS		
AE12	A6 00	LOBYTE	LDA A	0,X	
AE14	E6 01		LDA B	1,X	
AE16	58		ASL B		
AE17	49		ROL A		
AE18	58		ASL B		
AE19	49		ROL A		
AE1A	84 1F		AND A	#\$00011111	
AE1C	39		RTS		
AE1D	A6 00	HIBYTE	LDA A	0,X	
AE1F	44		LSR A		
AE20	44		LSR A		
AE21	44		LSR A		
AE22	8B 20		ADD A	\$\$20	ADD TAG
AE24	39		RTS		
AE25	86 1D	DARK	LDA A	\$\$1D	
AE27	7E FD 80		JMP	OUTCH	

SYNCHRONOUS DETECTOR
FOR IMAGE ALIGNER/PLOTTER

** THIS VERSION INTERLEAVES I & Q SAMPLES
 * ENTRY: MUX ADDRESS (\$DFF9) SET TO DESIRED MODULE
 * EXIT: VALUES FROM EVEN MUX ADDRESS IN [(0,X)(1,X)]
 * VALUE FROM ODD MUX ADDRESS IN [(2,X)(3,X)]

EE12	FOSTRAN	EQU	PIA+\$12	
EE22	NEGTRAN	EQU	PIA+\$22	
AE2A 6F 00	SYNDIET	CLR	0,X	
AE2C 6F 01		CLR	1,X	
AE2E 6F 02		CLR	2,X	
AE30 6F 03		CLR	3,X	
AE32 C6 02		LDA B	#2	
AE34 37	LOOP60	PSH B		
AE35 7D EE 11		TST	FOSTRAN-1	CLEAR CRB-7
AE38 7D EE 21		TST	NEGTRAN-1	
AE3B 0F		SEI		
AE3C B6 EE 13	SYNDT1	LDA A	FOSTRAN+1	
AE3F BA EE 23		ORA A	NEGTRAN+1	
AE42 2A F8		BPL	SYNDT1	WAIT FOR 60HZ CYCLE
AE44 C6 04		LDA B	#4	
AE46 37	SYNDT2	PSH B		
AE47 C6 02	BIGLOOP	LDA B	#2	
AE49 37	SIGNTST	PSH B		
AE4A 7D EE 10		TST	FOSTRAN-2	
AE4D 7D EE 20		TST	NEGTRAN-2	
AE50 7D ED 47		TST	SIGN	TAKE OTHER HALF CYCLE
AE53 27 0A		BEQ	NEGTST	
AE55 B6 EE 12	POSTST	LDA A	FOSTRAN	
AE58 2A FB		BPL	POSTST	
AE5A 7F ED 47		CLR	SIGN	
AE5D 20 0A		BRA	SAMPLE	
AE5F B6 EE 22	NEGTST	LDA A	NEGTRAN	
AE62 2A FB		BPL	NEGTST	
AE64 86 FF		LDA A	#\$FF	
AE66 B7 ED 47		STA A	SIGN	
AE69 C6 14	SAMPLE	LDA B	#20	
AE6B 5A	DELAY2	DEC B		
AE6C 26 FD		BNE	DELAY2	SIGNAL PHASE COMPENSATION
AE6E B7 DF FA		STA A	DAS+\$A	CONVERT COMMAND
AE71 B6 DF FB	WAIT	LDA A	DAS+\$B	FOR EOC
AE74 2A FB		BPL	WAIT	
AE76 B6 DF FD		LDA A	DAS+\$D	GET HI BYTE
AE79 F6 DF FC		LDA B	DAS+\$C	GET LO BYTE
AE7C 7D ED 47		TST	SIGN	
AE7F 26 05		BNE	SUBTR	
AE81 43		COM A		RE-INVERT
AE82 53		COM B		
AE83 0C		CLC		
AE84 20 01		BRA	ADDATA	
AE86 0D	SUBTR	SEC		DATA IS ALREADY INVERTED
AE87 E9 01		ADC B	1,X	
AE89 A9 00		ADC A	0,X	
AE8B 28 0D		BVC	STORE	NO OVERFLOW
AE8D 2A 08		BPL	MINUS	FILL FILTER WITH MAX. VALUE
AE8F C6 FF		LDA B	#\$FF	
AE91 86 7F		LDA A	#\$7F	

SYNCHRONOUS DETECTOR
FOR IMAGE ALIGNER/PLOTTER

AE93	20 05		BRA	STORE	
AE95	20 9D	LOOP61	BRA	LOOP60	BRANCH EXTENSION
AE97	5F	MINUS	CLR B		
AE98	86 80		LDA A	#\$80	
AE9A	E7 01	STORE	STA B	1,X	
AE9C	A7 00		STA A	0,X	
AE9E	33		PUL B		
AE9F	5A		DEC B		
AEA0	26 A7		BNE	SIGNTST	NEXT HALF-CYCLE
AEA2	B6 DF F9		LDA A	DAS+9	
AEA5	44		LSR A		
AEA6	25 07		RCS	ODD	
AEA8	7C DF F9		INC	DAS+9	OTHER CHANNEL
AEA9	08		INX		
AEAC	08		INX		
AEAD	20 98		BRA	BIGLOOP	
AEAF	7A DF F9	ODD	DEC	DAS+9	FIRST CHANNEL
AEB2	09		DEX		
AEB3	09		DEX		
AEB4	33		PUL B		
AEB5	5A		DEC B		
AEB6	26 8E		BNE	SYN0T2	TAKE ANOTHER CYCLE
AEB8	33		PUL B		
AEB9	0E		CLI		
AEBA	5A		DEC B		
AEBB	26 DB		BNE	LOOP61	
AEBD	39		RTS		

FOR IMAGE ALIGNER/PLOTTER

			*	WRITES CORRELATION COEFFICIENT ON CRT
			*	TEXT FOR HEADINGS
AE8E	43 4F 52	HEADG1	FCC	'CORRELATION COEF'
AECE	46 49 43		FCC	'FICIENT', \$D,\$A,\$A,\$A+\$80
AED9	4D 41 47	HEADG2	FCC	'MAGNITUDE PHAS'
AEE9	45 20 28		FCC	'E (Des)', \$D,\$A,\$A,7+\$80
AEF4	20 30 AE	DATA	FCC	'0', '.', '+\$80'
AEF7	07	CORCRT	TPA	
AEF8	36		PSH A	
AEF9	BD AB F4		JSR	PAGE
AEFC	C6 0F		LDA B	\$15
AEFE	BD FD A9	LINE	JSR	CRLF1
AF01	5A		DEC B	
AF02	26 FA		BNE	LINE MOVE DOWN 15 LINES
AF04	CE AE BE		LDX	\$HEADG1
AF07	8D 2F		BSR	SPSTR 1ST LINE OF TEXT
AF09	8D 2D		BSR	SPSTR 2ND LINE
AF0B	8D 2B		BSR	SPSTR 3RD LINE
AF0D	CE 18 24		LDX	\$COR1
AF10	BD FD 7A		JSR	OUT4HS PRINT MAGNITUDE
AF13	C6 05		LDA B	\$5
AF15	8D 28		BSR	SPACE
AF17	32		PUL A	GET SIGN OF ANGLE
AF18	84 01		AND A	\$1
AF1A	48		ASL A	
AF1B	8B 2B		ADD A	+\$2B
AF1D	BD FD 80		JSR	OUTCH
AF20	BD FD 8F		JSR	THB
AF23	A6 00	CRT1	LDA A	0,X
AF25	36		PSH A	
AF26	44		LSR A	
AF27	44		LSR A	
AF28	44		LSR A	
AF29	44		LSR A	
AF2A	BD FD 9C		JSR	THB1
AF2D	86 2E		LDA A	?"."
AF2F	BD FD 80		JSR	OUTCH
AF32	32		PUL A	
AF33	84 0F		AND A	+\$F
AF35	7E FD 9C		JMP	THB1
				RETURNS VIA RTS
AF38	C6 17	SPSTR	LDA B	\$23
AF3A	8D 03	SFSTR1	BSR	SPACE
AF3C	7E FD FF		JMP	OUTSTR
AF3F	BD FD 7E	SPACE	JSR	OUTSP
AF42	5A		DEC B	
AF43	26 FA		BNE	SPACE
AF45	39			RTS

GRAPHIC ALIGNMENT DISPLAY
FOR IMAGE ALIGNER/PLOTTER

* EXTERNAL REFERENCES

ED15	CALIBR	EQU	\$ED15
ED1E	XCOUNT	EQU	\$ED1E
ED23	YCOUNT	EQU	\$ED23
A820	XSTEP	EQU	\$A820
ED4D	MAG	EQU	\$ED4D
ED4E	MAG1	EQU	\$ED4E

* VECTORS FOR TEK 4006-1

AF46 1D 2C 66	AXES	FCB	\$1D, 'f @', 'f?_ ', \$1D, '00', '81', '@'+\$80
AF56 1D 2C 63	SMALSQ	FCB	\$1D, 'c/]', 'i]', 'iOC', 'cC', 'c/]', \$1D+\$80
AF66 1D 2A 74	MAGSQ	FCB	\$1D, '*t.N', '-xN', 'x1R', '*tr'
AF74 74 2E 4E		FCB	't.N', \$1D+\$80

AF78 BD AB F4	CROSS	JSR	PAGE
AF78 CE AF 46		LDX	*AXES
AF7E 7E FD FF	SQUARE	JMP	OUTSTR

AF81 7F ED 4E	GRAPH1	CLR	MAG1	
AF84 8D F2		BSR	CROSS	PAINT TARGET
AF86 8D F6		BSR	SQUARE	
AF88 B6 ED 17	GRAPH2	LDA A	PROGNR	
AF8B 81 0D		CMP A	#\$D	
AF8D 26 01		BNE	GO	
AF8F 39		RTS		SPURIOUS ENTRY FROM 'D' COMI
AF90 7F ED 4D	GO	CLR	MAG	
AF93 FE ED 21		LDX	XCOUNT+3	
AF96 8C 20 00		CPX	#512*16	10 BITS MAX. FOR CRT
AF99 2B 03		BMI	TEST2	
AF9B CE 20 00		LDX	#512*16	
AF9E 8C E0 00	TEST2	CPX	#-8192	
AFA1 2A 03		BPL	STORE1	
AFA3 CE E0 00		LDX	#-8192	
AFA6 FF ED 35	STORE1	STX	XCOORD	
AFA9 8C 02 00		CPX	#512	TOO BIG AFTER 16X MAG?
AFAC 2A 08		BPL	Y	
AFAE 8C FE 00		CPX	#-512	
AFB1 2B 03		BMI	Y	
AFB3 7C ED 4D		INC	MAG	O.K. SO FAR
AFB6 FE ED 26	Y	LDX	YCOUNT+3	
AFB9 8C 1E 00		CPX	#512*16-512	CRT VERTICAL IS SMALLER
AFBC 2B 03		BMI	TEST3	
AFBE CE 1E 00		LDX	#512*16-512	
AFC1 8C DF 00	TEST3	CPX	#-8192-256	
AFC4 2A 03		BPL	STORE2	
AFC6 CE DF 00		LDX	#-8192-256	
AFC9 FF ED 37	STORE2	STX	YCOORD	
AFCC 7D ED 4E		TST	MAG1	
AFCF 26 2B		BNE	CENTER	HAVE MAGNIFIED TARGET ALREA
AFD1 8C 01 00		CPX	#256	
AFD4 2A 16		BPL	SCALE	
AFD6 8C FF 00		CPX	#-256	
AFD9 2B 11		BMI	SCALE	
AFDB 7D ED 4D		TST	MAG	WAS X O.K.?

GRAPHIC ALIGNMENT DISPLAY
FOR IMAGE ALIGNER/PLOTTER

AFDE	27 0C		BEQ	SCALE	
AFE0	8D 96	MAG16X	BSR	CROSS	PAINT MAGNIFIED TARGET
AFE2	CE AF 66		LDX	#MAGSQ	
AFES	8D 97		BSR	SQUARE	
AFE7	7C ED 4E		INC	MAG1	
AFFA	20 10		BRA	CENTER	
AFEC	CE ED 35	SCALE	LDX	#XCOORD	REDUCE COORDINATES TO 1/16
AEFF	C6 04		LDA B	#4	
AFF1	67 00	LOOP	ASR	0,X	
AFF3	66 01		ROR	1,X	
AFF5	67 02		ASR	2,X	
AFF7	66 03		ROR	3,X	
AFF9	5A		DEC B		
AFFA	26 F5		BNE	LOOP	
AFFC	CE ED 35	CENTER	LDX	#XCOORD	
AFFF	A6 00		LDA A	0,X	
B001	8B 02		ADD A	\$512/256	SHIFT TO HORIZ. CENTER
B003	E6 01		LDA B	1,X	
B005	8D 0F		BSR	EXCHNG	
B007	08		INX		
B008	08		INX		
B009	E6 01		LDA B	1,X	
B00R	A6 00		LDA A	0,X	
B00D	CR 66		ADD B	\$390-32-256	SHIFT TO VERTICAL CENTER
B00F	89 01		ADC A	\$390-32/256	"VECTOR" ADDS BACK 32
B011	8D 03		BSR	EXCHNG	
B013	7E AD F2		JMP	VECTOR	
B016	47	EXCHNG	ASR A		
B017	56		ROR B		
B018	46		ROR A		
B019	56		ROR B		
B01A	46		ROR A		
B01B	A7 01		STA A	1,X	(A) NOW HAS L.S. 2 OF 10 BIT
B01D	E7 00		STA B	0,X	(B) HAS M.S. 8 OF 10 BITS
B01F	39	RETRN4	RTS		
B020			END		

STATEMENTS =574

FREE BYTES =19

NO ERRORS DETECTED

MEASURE

* SUBROUTINE MEASURE
 * MEASURES DEFT DEVICE OUTPUTS AND COMPUTES
 * VARIABLES FOR CORRELATION AND LEAST SQUARES ESTIMATI
 * MODE=1 FOR REFERENCE
 * MODE=2 FOR CORRELATE
 * MODE=3 FOR LEAST SQUARES
 * 1>=L>=NBDRY SETS FREQUENCY FLANE BOUNDARY FOR
 * LEAST SQUARES

1800		ORG	\$1800
1800	TEMP1	RMB	1
1801	TEMP2	RMB	1
1802	TEMP3	RMB	1
1803	TEMP4	RMB	1
1804	TEMP5	RMB	1
1805	TEMP6	RMB	1
1806	BCD1	RMB	1
1807	BCD2	RMB	1
1808	CTR	RMB	1
1809	CTR1	RMB	1
1814		ORG	\$1814
1814	X0ZERO	RMB	4
1818	Y0ZERO	RMB	4
181C	X1ZERO	RMB	4
1820	Y1ZERO	RMB	4
1824	COR1	RMB	1
1825	COR2	RMB	1
1826	COR3	RMB	1
1827	COR4	RMB	3
182A	COR9	RMB	1
182B	COR10	RMB	1
182C	I	RMB	1
182D	J	RMB	1
182E	ID	RMB	1
182F	JD	RMB	1
1830	UI1	RMB	1
1831	UI2	RMB	1
1832	VJ1	RMB	1
1833	VJ2	RMB	1
1834	MODE	RMB	1
1835	DTIME	RMB	1
1836	JSTART	RMB	1
1837	BDRY	RMB	1
1838	BDRYD	RMB	1
1839	NBDRY	RMB	1
183A	BDRYPT	RMB	1
183B	FIRST	RMB	1
183C	SENSOR	RMB	1
183D	NSAMP	RMB	1
183E	LOGNS	RMB	1
183F	BCDPTR	RMB	2
1841	L	RMB	1
1842	LIMIT	RMB	1
1843	X	RMB	1
1844	Y	RMB	1
1845	ZERO	RMB	4
1849	X1INC	RMB	2
1848	Y1INC	RMB	2

MEASURE

184D	NPT	RMB	1
184E	STACK1	RMB	1
184F	STACK2	RMB	1
1850	FUSHST	RMB	1
1851	THR11	RMB	1
1852	THR12	RMB	1
1853	THR21	RMB	1
1854	THR22	RMB	1
1855	HOW	RMB	1
1856	IJPTR	RMB	1
1857	IJPTR1	RMB	1
1858	ADRSW	RMB	1
1859	SETUP	RMB	1
185A	FINE	RMB	1
185B	K\$G	RMB	1
185C	KKKK	RMB	4
1860	CORSE	RMB	2
1862	ST	RMB	2
1864	SAVESF	RMB	2
1866	MAX	RMB	4
186A	SUM	RMB	4
186E	XTRAN	RMB	2
1870	YTRAN	RMB	2
1872	INELX	RMB	4
1876	DELY	RMB	4
187A	UP	RMB	4
187E	UP	RMB	4
1882	UV	RMB	4
1886	VV	RMB	4
188A	UU	RMB	4
188E	STEPS	RMB	2
1890	ANGLE	RMB	2
1892	SMAG	RMB	4
1896	SAVEX	RMB	2
1898	SAVEY	RMB	2
189A	DIRECT	RMB	1
189B	DTHR11	RMB	1
189C	DTHR12	RMB	1
189D	DTHR21	RMB	1
189E	DTHR22	RMB	1
189F	MPLX	RMB	2
18A1	SIGNI	RMB	1
18A2	SIGNQ	RMB	1
18A3	SFLAG	RMB	1
18A4	SNFLAG	RMB	1
18D8	ORG		\$18D8
18D8	DELTA	RMB	10D
18E2	BCDAR	RMB	512D
1AE2	BDRYLF	RMB	5
1AE7	BDRYCT	RMB	5
1AEC	S1VP1M	RMB	20D
1B00	S1UP1M	RMB	20D
1B14	S1UV1M	RMB	20D
1B28	S1V1M	RMB	20D
1B3C	S1U1M	RMB	20D
1B50	IJ	RMB	400D
1CEO	PS1MS	RMB	400D

MEASURE

1E70	MS1MS	RMB	400D	
BB68	PULL4	EQU	\$BB68	
BB8E	PUSH44	EQU	\$BB8E	
BBB4	PUSH41	EQU	\$BBB4	
BBD1	PUSH42	EQU	\$BBD1	
B9A0	DELAY1	EQU	\$B9A0	
BB17	BI\$BCD	EQU	\$BB17	
BBF4	TUNE	EQU	\$BBF4	
BA92	READ	EQU	\$BA92	
BC0D	RDDEFT	EQU	\$BC0D	
B9B6	ADDRES	EQU	\$B9B6	
8000	MATH	EQU	\$8000	
BDD6	CORDIC	EQU	\$BDD6	
BD28	GSRCH	EQU	\$BD28	
A65C	INDEX1	EQU	\$A65C	
A685	UPDATE	EQU	\$A685	
A63D	IDEX	EQU	\$A63D	
A74C	PHASDF	EQU	\$A74C	
*				
B040		ORG	\$B040	
B040	7F 18 3B	MEASRE	CLR	FIRST
B043	B6 18 34		LDAA	MODE WHICH MODE?
B046	4A		DECA	
B047	27 05		BEQ	LM1
B049	4A		DECA	
B04A	27 2D		BEQ	LM4
B04C	20 3C		BRA	LM5
B04E	7F 18 2C	LM1	CLR	I MODE 1: INITIALIZE
B051	B6 01		LDAA	#01
B053	B7 18 43		STAA	X
B056	B7 18 37		STAA	BDRY
B059	B6 18 36		LDAA	JSTART
B05C	B7 18 2D		STAA	J
B05F	CE 00 00		LDX	#0
B062	BF 18 00		STS	TEMP1
B065	8E 1B 4F		LDS	#IJ-1 S1U(5) LS
B068	4F		CLRA	
B069	36	LM2	PSHA	CLEAR S1 VARIABLES
B06A	08		INX	
B06B	8C 00 3C		CPX	\$60D
B06E	26 F9		BNE	LM2
B070	BE 18 00	LM3	LDS	TEMP1
B073	7F 18 3C		CLR	SENSOR O=REFERENCE SENSOR
B076	7E B1 28		JMP	TO ADDRES
B079	CE 00 00	LM4	LDX	#0 MODE 2: INITIALIZE
B07C	FF 18 6A		STX	SUM
B07F	FF 18 6C		STX	SUM+2
B082	FF 18 92		STX	SMAG
B085	FF 18 94		STX	SMAG+2
B088	20 11		BRA	LM7
B08A	86 01	LM5	LDAA	#1 MODE 3: INITIALIZE
B08C	B7 18 37		STAA	BDRY
B08F	CE 1A EC		LDX	#S1VP1M
B092	4F		CLRA	
B093	A7 00	LM6	STAA	0,X CLEAR S1UF AND S1VP
B095	08		INX	
B096	8C 1B 14		CPX	#S1UV1M

MEASURE

B099	26 F8		BNE	LM6	
B09B	86 01	LM7	LDAA	#1	MODES 2 AND 3: INITIALIZE
B09D	B7 18 44		STAA	Y	
B0A0	CE 18 DE		LDX	#BCDAR-4	
B0A3	FF 18 3F		STX	BCDFTR	
B0A6	CE 18 50		LDX	#IJ	
B0A9	FF 18 56		STX	IJPTR	
B0AC	A6 00		LDAA	0,X	
B0AE	B7 18 2D		STAA	J	PASS IU(1), JV(1) TO ADDRESS
B0B1	CE 1C 18		LDX	#IJ+200D	
B0B4	A6 00		LDAA	0,X	
B0B6	B7 18 2C		STAA	I	
B0B9	86 01		LDAA	#1	
B0BB	B7 18 3C		STAA	SENSOR	1=ALIGNING SENSOR
B0BE	7E B1 28		JMP	LM8	TO TUNE
B0C1	FE 18 2C	LM9	LDX	I	SAVE OLD VALUES
B0C4	FF 18 2E		STX	ID	
B0C7	B6 18 34		LDAA	MODE	WHICH MODE?
B0CA	4A		DECA		
B0CB	27 0A		BEQ	LM10	
B0CD	4A		DECA		
B0CE	27 34		BEQ	LM131	
B0D0	7D 18 A4		TST	SNFLAG	
B0D3	27 34		BEQ	LM14	
B0D5	20 28		BRA	LM130	
B0D7	B6 18 37	LM10	LDAA	BDRY	MODE 1: SET UP
B0DA	B7 18 38		STAA	BDRYD	
B0DD	BD A6 5C		JSR	INDEX1	
B0E0	7D 18 2C		TST	I	
B0E3	26 18		BNE	LM13	
B0E5	B6 18 2D		LDAA	J	
B0E8	CE 1A E2		LDX	#BDRYLF	TABLE OF NORMALIZED BOUNDARIES
B0EB	5F		CLRB		
B0EC	A1 00	LM11	CMPA	0,X	
B0EE	27 09		BEQ	LM12	
B0F0	5C		INC B		
B0F1	08		INX		
B0F2	F1 18 39		CMPB	NBDRY	
B0F5	27 06		BEQ	LM13	IF ZERO, IN OLD REGION
B0F7	20 F3		BRA	LM11	
B0F9	5C	LM12	INC B		IN NEW REGION
B0FA	F7 18 37		STAB	BDRY	
B0FD	20 26	LM13	BRA	LM15	
B0FF	BD BC 0D	LM130	JSR	RDDEFT	
B102	20 08		BRA	LM141	
B104	7D 18 58	LM131	TST	ADRSW	
B107	2E F6		BGT	LM130	
B109	BD BA 92	LM14	JSR	READ	MODE 2 AND 3: SET UP
B10C	FE 18 56	LM141	LDX	IJPTR	
B10F	08		INX		
B110	FF 18 56		STX	IJPTR	
B113	A6 00		LDAA	0,X	
B115	B7 18 2D		STAA	J	J=IJ(IJPTR)
B118	4F		CLRA		
B119	C6 C8		LDAB	#200D	
B11B	BD A6 3D		JSR	INDEX	
B11E	A6 00		LDAA	0,X	

MEASURE

B120	B7 18 2C	STAA	I	I=IJ(IJPTR+200)
B123	20 03	BRA	LM8	
B125	BD BC 0D	LM15	JSR	RDDEFT
B128	BD B9 B6	LM8	JSR	ADDRES
B12B	BD BR F4	LM16	JSR	TUNE
B12E	7D 18 3B		TST	FIRST
B131	26 09		BNE	LM17
B133	RD B9 A0		JSR	DELAY1
B136	7C 18 3B		INC	FIRST
B139	7E B0 C1		JMP	LM9
B13C	7F 18 2A	LM17	CLR	CORDIC VECTOR MODE
B13F	7F 18 2B		CLR	COR10
B142	BD BD D6		JSR	CORDIC
B145	B6 18 34		LDAA	MODE
B148	4A		DECA	
B149	27 09		BEQ	LM18
B14B	4A		DECA	
B14C	27 03		BEQ	LM171
B14E	7E B3 1A		JMP	LM22
B151	7E B2 12	LM171	JMP	LM29
B154	7D 18 A3	LM18	TST	SFLAG
				MODE 1: SEARCH TRANSFORM
B157	2E 06		BGT	LM180
B159	BD A6 85		JSR	UPDATE
B15C	7E B2 94		JMP	LM20
B15F	B6 18 24	LM180	LDAA	COR1
B162	F6 18 25		LDAB	COR2
B165	F0 18 52		SUBB	THR12
B168	B2 18 51		SBCA	THR11
B16B	2C 03		BGE	LM19
B16D	7E B2 94		JMP	LM20
				REJECT: INSIGNIFICANT MAG
B170	B6 18 2C	LM19	LDAA	I
B173	B7 18 8B		STAA	UU+1
B176	B6 18 2D		LDAA	J
B179	B7 18 87		STAA	UU+1
B17C	7F 18 66		CLR	MAX
B17F	7F 18 67		CLR	MAX+1
B182	B6 02		LDAA	#2
B184	B7 18 62		STAA	ST
B187	BD BD 28		JSR	GSRCH
B18A	7A 18 62		DEC	ST
B18D	BD BD 28		JSR	GSRCH
B190	B6 18 8B		LDAA	UU+1
B193	B7 18 2C		STAA	I
B196	B6 18 87		LDAA	UU+1
B199	B7 18 2D		STAA	J
B19C	BD B9 B6		JSR	ADDRES
B19F	BD BB F4		JSR	TUNE
B1A2	B6 18 66		LDAA	MAX
B1A5	F6 18 67		LDAB	MAX+1
B1A8	F0 18 52		SUBB	THR12
B1AB	B2 18 51		SBCA	THR11
B1AE	2C 03		BGE	LM190
B1B0	7E B2 94		JMP	LM20
B1B3	FE 18 66	LM190	LDX	MAX
B1B6	FF 18 24		STX	COR1
B1B9	FE 18 68		LDX	MAX+2
B1BC	FF 18 2A		STX	COR9

B1BF	CE 1C 17	LDX	#MS1MS-601D	
B1C2	FF 18 00	STX	TEMP1	
B1C5	B6 18 00	LDAA	TEMP1	
B1C8	F6 18 01	LDAB	TEMP2	
B1CB	FB 18 43	ADDB	X	
B1CE	89 00	ADCA	\$0	
B1D0	B7 18 00	STAA	TEMP1	
B1D3	F7 18 01	STAB	TEMP2	
B1D6	FE 18 00	LDX	TEMP1	
B1D9	B6 18 2E	LDAA	ID	
B1DC	A7 00	STAA	0,X	IJ(X+200)=ID
B1DE	B6 18 00	LDAA	TEMP1	
B1E1	C0 C8	SUBB	\$200D	
B1E3	82 00	SBCA	\$0	
B1E5	B7 18 00	STAA	TEMP1	
B1E8	F7 18 01	STAB	TEMP2	
B1EB	FE 18 00	LDX	TEMP1	
B1EE	B6 18 2F	LDAA	JD	
B1F1	A7 00	STAA	0,X	IJ(X)=JD
B1F3	F6 18 38	LDAB	BDRYD	UPDATE S1 VARIABLES
B1F6	58	ASLB		
B1F7	58	ASLB		
B1F8	C0 03	SUBB	\$3	
B1FA	F7 18 3A	STAB	BDRYPT	POINTS TO MS BYTE
B1FD	F6 18 2E	LDAB	ID	S1U(BDRYD)=S1U(BDRYD)+ID*ID
B200	4F	CLRA		
B201	BD BB D1	JSR	PUSH42	
B204	F6 18 2E	LDAB	ID	
B207	4F	CLRA		
B208	BD BB D1	JSR	PUSH42	
B20B	86 0A	LDAA	\$10D	
B20D	BD 80 00	JSR	MATH	
B210	20 03	BRA	LM28	
B212	7E B2 A6	LM29	JMP	LM21
B215	CE 1B 3B	LM28	LDX	#S1U1M-1
B218	B6 18 3A	LDAA	BDRYPT	
B21B	BD BB 8E	JSR	PUSH44	
B21E	86 0B	LDAA	\$11D	
B220	BD 80 00	JSR	MATH	
B223	CE 1B 3B	LDX	#S1U1M-1	
B226	B6 18 3A	LDAA	BDRYPT	
B229	BD BB 68	JSR	PULL4	
B22C	B6 18 2F	LDAA	JD	S1V(BDRYD)=S1V(BDRYD)+JD*JD
B22F	BD BB B4	JSR	PUSH41	
B232	B6 18 2F	LDAA	JD	
B235	BD BB B4	JSR	PUSH41	
B238	86 0A	LDAA	\$10D	
B23A	BD 80 00	JSR	MATH	
B23D	CE 1B 27	LDX	#S1V1M-1	
B240	B6 18 3A	LDAA	BDRYPT	
B243	BD BB 8E	JSR	PUSH44	
B246	86 0B	LDAA	\$11D	
B248	BD 80 00	JSR	MATH	
B24B	CE 1B 27	LDX	#S1V1M-1	
B24E	B6 18 3A	LDAA	BDRYPT	
B251	BD BB 68	JSR	PULL4	
B254	F6 18 2E	LDAB	ID	S1UV(BDRYD)=S1UV(BDRYD)+ID*J

MEASURE

B257	4F	CLRA		
B258	BD BB D1	JSR	PUSH42	
B258	B6 18 2F	LDAA	JD	
B25E	BD BB B4	JSR	PUSH41	
B261	B6 0A	LDAA	\$10D	
B263	BD 80 00	JSR	MATH	
B266	CE 1B 13	LDX	*S1UV1M-1	
B269	B6 18 3A	LDAA	BDRYPT	
B26C	BD BB BE	JSR	PUSH44	
B26F	B6 0B	LDAA	\$11D	
B271	BD 80 00	JSR	MATH	
B274	CE 1B 13	LDX	*S1UV1M-1	
B277	B6 18 3A	LDAA	BDRYPT	
B27A	BD BB 68	JSR	PULL4	
B27D	CE 1A E6	LDX	*BDRYCT-1	BDRYCT(BDRYD)=X
B280	F6 18 38	LDAB	BDRYD	
B283	4F	CLRA		
B284	BD A6 3D	JSR	IDEX	
B287	B6 18 43	LDAA	X	
B28A	A7 00	STAA	0,X	
B28C	B0 18 4D	SUBA	NPT	
B28F	2C 14	BGE	LM201	GOT NPT SIGNIFICANT POINTS
B291	7C 18 43	INC	X	X=X+1
B294	F6 18 2C	LM20	LDAB	I
B297	4F	CLRA		
B298	F0 18 42	SUBB	LIMIT	
B29B	82 00	SBCA	#0	
B29D	2C 03	BGE	LM202	
B29F	7E B3 CF	JMP	LM26	DO ANOTHER POINT
B2A2	7A 18 43	LM202	DEC	X
B2A5	39	LM201	RTS	
B2A6	CE 1E 6E	LM21	LDX	*MS1MS-2
B2A9	FF 18 00	STX	TEMP1	MODE 2: CORRELATION
B2AC	B6 18 00	LDAA	TEMP1	
B2AF	F6 18 01	LDAB	TEMP2	
B2B2	FB 18 44	ADDB	Y	
B2B5	89 00	ADCA	#0	
B2B7	FB 18 44	ADDB	Y	
B2BA	89 00	ADCA	#0	
B2BC	B7 18 00	STAA	TEMP1	
B2BF	F7 18 01	STAB	TEMP2	
B2C2	FE 18 00	LDX	TEMP1	
B2C5	A8 00	LDAA	0,X	GET MS(Y)
B2C7	E6 01	LDAB	1,X	
B2C9	BD BB D1	JSR	PUSH42	
B2CC	B6 18 24	LDAA	COR1	
B2CF	F6 18 25	LDAB	COR2	
B2D2	BD BB D1	JSR	PUSH42	
B2D5	B6 0A	LDAA	\$10D	
B2D7	BD 80 00	JSR	MATH	MS(Y)*MAG
B2DA	CE 18 6A	LDX	*SUM	
B2DD	4F	CLRA		
B2DE	BD BB BE	JSR	PUSH44	
B2E1	B6 0B	LDAA	\$11D	
B2E3	BD 80 00	JSR	MATH	SUM=SUM+MAG*MS(Y)
B2E6	CE 18 6A	LDX	*SUM	
B2E9	4F	CLRA		

ASURE

B2EA	BD BB 68	JSR	PULL4	
B2ED	B6 18 24	LDAA	COR1	
B2F0	F6 18 25	LDAB	COR2	
B2F3	BD BB D1	JSR	PUSH42	
B2F6	B6 18 24	LDAA	COR1	
B2F9	F6 18 25	LDAB	COR2	
B2FC	BD BB D1	JSR	PUSH42	
B2FF	86 0A	LDAA	\$10D	
B301	BD 80 00	JSR	MATH	MAG*MAG
B304	CE 18 92	LDX	*SMAG	
B307	4F	CLRA		
B308	BD BB 8E	JSR	PUSH44	
B30B	86 0B	LDAA	\$11D	
B30D	BD 80 00	JSR	MATH	SMAG=SMAG+MAG*MAG
B310	CE 18 92	LDX	*SMAG	
B313	4F	CLRA		
B314	BD BB 68	JSR	PULL4	
B317	7E B3 D6	JMP	LM30	
B31A	CE 1A E6	LM22	LDX	#BDRYCT-1 MODE 3: LEAST SQUARE
B31D	F6 18 37	LDAB	BDRY	
B320	4F	CLRA		
B321	BD A6 3D	JSR	IDEX	
B324	A6 00	LDAA	0,X	
B326	B0 18 44	SUBA	Y	
B329	2C 05	BGE	LM23	
B32B	7C 18 37	INC	BDRY	
B32E	20 EA	BRA	LM22	
B330	B6 18 37	LM23	LDAA	BDRY
B333	B0 18 41	SUBA	L	
B336	2F 04	BLE	LM24	
B338	7A 18 44	DEC	Y	
B33B	39	RTS		BDRY>L
B33C	F6 18 37	LM24	LDAB	BDRY
B33F	58	ASLB		
B340	58	ASLB		
B341	C0 03	SUBB	#3	
B343	F7 18 3A	STAB	BDRYPT	
B346	CE 1C DE	LDX	#PS1MS-2	
B349	F6 18 44	LDAB	Y	
B34C	4F	CLRA		
B34D	58	ASLB		
B34E	49	ROLA		
B34F	BD A6 3D	JSR	IDEX	
B352	20 03	BRA	LM243	
B354	7E B3 C4	LM240	JMP	LM25
B357	EE 00	LM243	LDX	0,X
B359	FF 18 02	STX	TEMP3	
B35C	FE 18 2A	LDX	COR9	
B35F	FF 18 00	STX	TEMP1	
B362	CE 18 00	LDX	*TEMP1	
B365	BD A7 4C	JSR	PHASDF	
B368	EE 00	LDX	0,X	
B36A	FF 18 2A	STX	COR9	
B36D	F6 18 2E	LDAB	ID	S1UP(BDRY)=S1UP(BDRY)+ID*FD
B370	4F	CLRA		
B371	BD BB D1	JSR	PUSH42	
B374	B6 18 2A	LDAA	COR9	

MEASURE

B377	F6 18 28		LDAB	COR10	
B37A	BD BB D1		JSR	PUSH42	
B37D	86 0A		LDAA	\$10D	
B37F	BD 80 00		JSR	MATH	
B382	CE 1A FF		LDX	\$S1VP1M-1	
B385	B6 18 3A		LDAA	BDRYPT	
B388	BD BB 8E		JSR	PUSH44	
B38B	86 0B		LDAA	\$11D	
B38D	BD 80 00		JSR	MATH	
B390	CE 1A FF		LDX	\$S1VP1M-1	
B393	B6 18 3A		LDAA	BDRYPT	
B396	BD BB 68		JSR	PULL4	
B399	B6 18 2F		LDAA	JD	$S1VP(BDRY)=S1VP(BDRY)+JD*FD$
B39C	BD BB B4		JSR	PUSH41	
B39F	B6 18 2A		LDAA	COR9	
B3A2	F6 18 28		LDAB	COR10	
B3A5	BD BB D1		JSR	PUSH42	
B3A8	86 0A		LDAA	\$10D	
B3AA	BD 80 00		JSR	MATH	
B3AD	CE 1A EB		LDX	\$S1VP1M-1	
B3B0	B6 18 3A		LDAA	BDRYPT	
B3B3	BD BB 8E		JSR	PUSH44	
B3B6	86 0B		LDAA	\$11D	
B3B8	BD 80 00		JSR	MATH	
B3BB	CE 1A EB		LDX	\$S1VP1M-1	
B3BE	B6 18 3A		LDAA	BDRYPT	
B3C1	BD BB 68		JSR	PULL4	
B3C4	7C 18 44	LM25	INC	Y	MODES 2 AND 3
B3C7	B6 18 43		LDAA	X	
B3CA	80 18 44		SUBA	Y	
B3CD	2D 06		BLT	LM27	NO MORE SIGNIFICANT POINTS
B3CF	BD B9 A0	LM26	JSR	DELAY1	
B3D2	7E B0 C1		JMP	LM9	NEXT POINT
B3D5	39	LM27	RTS		TO MAIN ALIGNMENT PROGRAM
B3D6	7C 18 44	LM30	INC	Y	MODE 2
B3D9	B6 18 43		LDAA	X	
B3DC	80 18 44		SUBA	Y	
B3DF	2C EE		BGE	LM26	
B3E1	CE 18 7E	LM31	LDX	\$VP+4	
B3E4	4F		CLRA		
B3E5	BD BB 8E		JSR	PUSH44	
B3E8	CE 18 7A		LDX	\$VP	
B3EB	4F		CLRA		
B3EC	BD BB 8E		JSR	PUSH44	MAX NFPN
B3EF	4F		CLRA		
B3F0	BD BB B4		JSR	PUSH41	
B3F3	CE 18 6A		LDX	\$SUM	
B3F6	4F		CLRA		
B3F7	BD BB 8E		JSR	PUSH44	
B3FA	86 07		LDAA	\$7	
B3FC	BD 80 00		JSR	MATH	SUM NFPN
B3FF	4F		CLRA		
B400	BD BB B4		JSR	PUSH41	
B403	CE 18 6A		LDX	\$SUM	
B406	4F		CLRA		
B407	BD BB 8E		JSR	PUSH44	
B40A	B6 07		LDAA	\$7	

MEASURE

B40C	BD 80 00	JSR	MATH	SUM NFPN
B40F	86 01	LDAA	#1	
B411	BD 80 00	JSR	MATH	SUM*SUM (FP)
B414	4F	CLRA		
B415	BD BB B4	JSR	PUSH41	
B418	CE 18 92	LDX	#SMAG	
B41B	4F	CLRA		
B41C	BD BB 8E	JSR	PUSH44	
B41F	86 07	LDAA	#7	
B421	BD 80 00	JSR	MATH	SMAG NFPN
B424	86 02	LDAA	#2	
B426	BD 80 00	JSR	MATH	SUM=SUM*SUM/SMAG
B429	CE 18 82	LDX	#UV	
B42C	4F	CLRA		
B42D	BD BB 68	JSR	PULL4	
B430	CE 18 86	LDX	#UV+4	
B433	4F	CLRA		
B434	BD BB 68	JSR	PULL4	UV=SUM
B437	CE 18 86	LDX	#UV+4	
B43A	4F	CLRA		
B43B	BD BB 8E	JSR	PUSH44	
B43E	CE 18 82	LDX	#UV	
B441	4F	CLRA		
B442	BD BB 8E	JSR	PUSH44	
B445	86 02	LDAA	#2	
B447	BD 80 00	JSR	MATH	MAX/SUM
B44A	CE 18 6A	LDX	#SUM	
B44D	4F	CLRA		
B44E	BD BB 68	JSR	PULL4	
B451	CE 18 6A	LDX	#SUM	
B454	4F	CLRA		
B455	BD BB 68	JSR	PULL4	IF SUM+3<0 THEN SUM>MAX
B458	39	RTS		
B459		END		

STATEMENTS =548

FREE BYTES =1023

NO ERRORS DETECTED

ALIGN

1800	ORG	\$1800
1800	TEMP1	RMB 1
1801	TEMP2	RMB 1
1802	TEMP3	RMB 1
1803	TEMP4	RMB 1
1804	TEMP5	RMB 1
1805	TEMP6	RMB 1
1806	BCD1	RMB 1
1807	BCD2	RMB 1
1808	CTR	RMB 1
1809	CTR1	RMB 1
1812	ORG	\$1812
1812	PHASE	RMB 2
1814	X0ZERO	RMB 4
1818	Y0ZERO	RMB 4
181C	X1ZERO	RMB 4
1820	Y1ZERO	RMB 4
1824	COR1	RMB 1
		REAL
1825	COR2	RMB 1
1826	COR3	RMB 1
1827	COR4	RMB 3
		IMAG
182A	COR9	RMB 1
182B	COR10	RMB 1
182C	I	RMB 1
182D	J	RMB 1
182E	ID	RMB 1
182F	JD	RMB 1
1830	UI1	RMB 1
1831	UI2	RMB 1
1832	VJ1	RMB 1
1833	VJ2	RMB 1
1834	MODE	RMB 1
1835	DTIME	RMB 1
1836	JSTART	RMB 1
1837	BDRY	RMB 1
1838	BDRYD	RMB 1
1839	NBDRY	RMB 1
183A	BDRYPT	RMB 1
183B	FIRST	RMB 1
183C	SENSOR	RMB 1
183D	NSAMP	RMB 1
183E	LOGS	RMB 1
183F	BCDPTR	RMB 2
1841	L	RMB 1
1842	LIMIT	RMB 1
1843	X	RMB 1
1844	Y	RMB 1
1845	ZERO	RMB 4
1849	X1INC	RMB 2
184B	Y1INC	RMB 2
184D	NPT	RMB 1
184E	STACK1	RMB 1
184F	STACK2	RMB 1
1850	PUSHST	RMB 1
1851	THR11	RMB 1
1852	THR12	RMB 1
1853	THR21	RMB 1

ALIGN

1854	THR22	RMB	1
1855	HOW	RMB	1
1856	IJFTR	RMB	1
1857	IJPTR1	RMB	1
1858	ADRSW	RMB	1
1859	SETUP	RMB	1
185A	FINE	RMB	1
185B	K\$G	RMB	1
185C	KKKK	RMB	4
1860	CORSE	RMB	2
1862	ST	RMB	2
1864	SAVESP	RMB	2
1866	MAX	RMB	4
186A	SUM	RMB	4
186E	XTRAN	RMB	2
1870	YTRAN	RMB	2
1872	DELX	RMB	4
1876	DELY	RMB	4
187A	VP	RMB	4
187E	UP	RMB	4
1882	UV	RMB	4
1886	UU	RMB	4
188A	STEPS	RMB	2
1890	ANGLE	RMB	2
1892	SMAG	RMB	4
1896	SAVEX	RMB	2
1898	SAVEY	RMB	2
189A	DIRECT	RMB	1
189B	DTHR11	RMB	1
189C	DTHR12	RMB	1
189D	DTHR21	RMB	1
189E	DTHR22	RMB	1
189F	MPLX	RMB	2
18A1	SIGNI	RMB	1
18A2	SIGNQ	RMB	1
18A3	SFLAG	RMB	1
18A4	SNFLAG	RMB	1
18D8	ORG		\$18D8
18D8	DELTA	RMB	10D
18E2	BCDAR	RMB	512D
1AE2	BDRLYLF	RMB	5
1AE7	BDRYCT	RMB	5
1AEC	S1VP1M	RMB	20D
1B00	S1UP1M	RMB	20D
1B14	S1UV1M	RMB	20D
1B28	S1V1M	RMB	20D
1B3C	S1U1M	RMB	20D
1B50	IJ	RMB	400D
1CE0	PS1MS	RMB	400D
1E70	MS1MS	RMB	400D
BB68	FULL4	EQU	\$BB68
BB8E	PUSH44	EQU	\$BB8E
BBB4	PUSH41	EQU	\$BBB4
BBD1	PUSH42	EQU	\$BBD1
A6D1	THRSET	EQU	\$A6D1
A9A1	SUPRVR	EQU	\$A9A1

ALIGN

8000	MATH	EQU	\$8000	
B040	MEASRE	EQU	\$B040	
A77F	PHSSET	EQU	\$A77F	
FE2D	FANTOM	EQU	\$FE2D	
A820	STPMTR	EQU	\$A820	
ED15	CALIBR	EQU	\$ED15	
AA31	ERROR	EQU	\$AA31	
AF81	GRAPH1	EQU	\$AF81	
ED17	PROGMR	EQU	\$ED17	
	*	SUBROUTINE		ALIGN
	*	MAIN ALIGNMENT PROGRAM		
B470		ORG	\$B470	
B470	7F 18 A4	INALGN	CLR	SNFLAG
B473	B6 ED 17		LDAA	PROGMR
B476	46		RORA	
B477	24 33		BCC	LA2
B479	7C 18 A4		INC	SNFLAG
B47C	20 2E		BRA	LA2
B47E	7D 18 A4	DOUBLE	TST	SNFLAG
B481	2E 20		BGT	H3
B483	8D 0B		BSR	HOWSET
B485	20 5C		BRA	LA21
B487	7D 18 A4	SINGLE	TST	SNFLAG
B48A	27 17		BEQ	H3
B48C	8D 02		BSR	HOWSET
B48E	20 69		BRA	LA23
B490	7F 18 55	HOWSET	CLR	HOW
B493	B6 ED 17		LDAA	PROGMR
B496	46		RORA	
B497	25 03		BCS	H1
B499	7C 18 55		INC	HOW
B49C	7D ED 15	H1	TST	CALIBR
B49F	26 07		BNE	H2
B4A1	31		INS	
B4A2	31		INS	
B4A3	31	H3	INS	
B4A4	31		INS	
B4A5	7E AA 31		JMP	ERROR
B4A8	BD AF 81	H2	JSR	GRAPH1
B4AB	39		RTS	
B4AC	86 01	LA2	LDAA	\$1
B4AE	B7 18 34		STAA	MODE
B4B1	B7 18 55		STAA	HOW
B4B4	7F 18 58		CLR	ADRSW
B4B7	7F 18 A3		CLR	SFLAG
B4BA	7D 18 A4		TST	SNFLAG
B4BD	27 18		BEQ	LA20
B4BF	FE 18 1C		LDX	X1ZERO
B4C2	FF 18 14		STX	X0ZERO
B4C5	FE 18 1E		LDX	X1ZERO+2
B4C8	FF 18 16		STX	X0ZERO+2
B4CB	FE 18 20		LDX	Y1ZERO
B4CE	FF 18 18		STX	Y0ZERO
B4D1	FE 18 22		LDX	Y1ZERO+2
B4D4	FF 18 1A		STX	Y0ZERO+2
B4D7	BD B0 40	LA20	JSR	MEASRE
B4DA	BD A6 D1		JSR	THRSET

CHARACTERIZE TRANSFORM
SET THRESHOLD

ALIGN

B4DD	7C 18 A3		INC	SFLAG	
B4E0	BD 80 40		JSR	MEASRE	REFERENCE
B4E3	86 02	LA21	LDA	\$2	ENTER TO READ SAME REF
B4E5	B7 18 34		STA	MODE	
B4E8	86 FF		LDA	\$\$FF	
B4EA	B7 18 58		STA	ADRSW	
B4ED	BD 80 40		JSR	MEASRE	STORE BCD ADDRESSES
B4F0	7D 18 A4		TST	SNFLAG	
B4F3	27 04		BREQ	LA23	
B4F5	7C E0 15		INC	CALIBR	
B4F8	39		RTS		
B4F9	7D 18 55	LA23	TST	HOW	
B4FC	2E 03		BGT	LA24	
B4FE	7E B5 9F		JMP	LA85	
B501	CE 00 00	LA24	LDX	\$0	ENTER TO SKIP REF
B504	FF 18 6E		STX	XTRAN	
B507	FF 18 70		STX	YTRAN	
B50A	FF 18 7A		STX	VP	
B50D	FF 18 7C		STX	VP+2	
B510	FF 18 7E		STX	VP+4	
B513	CE 00 80		LDX	\$\$0080	
B516	FF 18 80		STX	VP+6	FF ZERO
B519	4F		CLRA		
B51A	5F		CLRB		
B51B	F0 18 61		SUBB	CORSE+1	
B51E	B2 18 60		SRCA	CORSE	
B521	F7 18 91		STAB	ANGLE+1	
B524	B7 18 90		STA	ANGLE	
B527	F7 18 63		STAB	ST+1	
B52A	B7 18 62		STA	ST	
B52D	BD A8 20		JSR	STPMTR	-CORSE
B530	86 01		LDA	\$1	
B532	B7 18 58		STA	ADRSW	
B535	86 02		LDA	\$2	
B537	B7 18 34		STA	MODE	
B53A	BD 80 40	LA3	JSR	MEASRE	CORRELATE: CORSE LOOP
B53D	7D 18 6D		TST	SUM+3	
B540	2E 09		BGT	LA4	
B542	BD B8 1D	LA32	JSR	MAXSUM	SUM>MAX
B545	FE 18 62		LDX	ST	
B548	FF 18 8E		STX	STEPS	STEPS=ST
B54B	F6 18 63	LA4	LDA	ST+1	ST=ST+2 DEG
B54E	CB 14		ADDB	\$\$14	
B550	F7 18 63		STAB	ST+1	
B553	B6 18 62		LDA	ST	
B556	89 00		ADCA	\$0	
B558	B7 18 62		STA	ST	
B55B	F0 18 61		SUBB	CORSE+1	ST-CORSE-0.1 DEG
B55E	B2 18 60		SBCA	CORSE	
B561	C0 01		SUBB	\$1	
B563	82 00		SBCA	\$0	
B565	2C 0E		BGE	LA5	
B567	86 14		LDA	\$\$14	
B569	B7 18 91		STA	ANGLE+1	
B56C	7F 18 90		CLR	ANGLE	ANGLE=2 DEG
B56F	BD A8 20		JSR	STPMTR	
B572	7E B5 3A		JMP	LA3	TO MEASRE

ALIGN

B575	B6	18	8E	LA5	LDAA	STEPS	ST>CORSE
B578	F6	18	8F		LDAB	STEPS+1	
B579	F0	18	63		SUBB	ST+1	
B57E	B2	18	62		SRCA	ST	
B581	CB	14			ADDB	\$\$14	
B583	89	00			ADCA	0	
B585	B7	18	90		STAA	ANGLE	ANGLE=STEPS-ST+2 DEG
B588	F7	18	91		STAB	ANGLE+1	
B58B	BD	AB	20		JSR	STPMTR	ROTATE BACK
B58E	B6	18	5A		LDAA	FINE	
B591	B7	18	8F		STAA	STEPS+1	
B594	7F	18	8E		CLR	STEPS	
B597	BD	B7	B2	LA6	JSR	FINSCH	
B59A	77	18	8F		ASR	STEPS+1	
B59D	26	F8			BNE	LA6	
B59F	86	03		LA85	LDAA	#3	ENTER TO SKIP REF. ROTATION
B5A1	B7	18	34		STAA	MODE	
B5A4	86	01			LDAA	#1	
B5A6	B7	18	41		STAA	L	
B5A9	B7	18	58		STAA	ADRSW	
B5AC	B7	18	8F		STAA	STEPS+1	
B5AF	7F	18	8E		CLR	STEPS	
B5B2	86	F5			LDAA	\$\$F5	
B5B4	B7	18	5E		STAA	KKKK+2	
B5B7	BD	A7	7F	LA9	JSR	PHSSET	MEASURE PHASE DIFF
B5B8	86	01			LDAA	#1	
B5BC	B7	18	58		STAA	ADRSW	
B5BF	BD	B0	40		JSR	MEASRE	LEAST SQUARES
B5C2	7C	18	5E		INC	KKKK+2	
B5C5	B6	18	44		LDAA	Y	
B5C8	4A				DECA		
B5C9	2E	03			BGT	LA90	
B5CB	7E	B7	9E		JMP	LA12	
B5CE	B6	18	5F	LA90	LDAA	KKKK+3	
B5D1	B7	18	5C		STAA	KKKK	
B5D4	CE	1A	EB		LDX	\$\$S1UP1M-1	
B5D7	FF	18	00		STX	TEMP1	
B5DA	CE	18	7A		LDX	\$\$UP	
B5DD	FF	18	02		STX	TEMP3	
B5E0	BD	B8	49		JSR	SUMPAR	
B5E3	CE	1A	FF		LDX	\$\$S1UP1M-1	
B5E6	FF	18	00		STX	TEMP1	
B5E9	CE	18	7E		LDX	\$\$UP	
B5EC	FF	18	02		STX	TEMP3	
B5EF	BD	B8	49		JSR	SUMPAR	
B5F2	B6	18	5B		LDAA	K\$G	
B5F5	B7	18	5C		STAA	KKKK	
B5F8	CE	18	13		LDX	\$\$S1UU1M-1	
B5FB	FF	18	00		STX	TEMP1	
B5FE	CE	18	82		LDX	\$\$UV	
B601	FF	18	02		STX	TEMP3	
B604	BD	B8	49		JSR	SUMPAR	
B607	CE	18	27		LDX	\$\$S1V1M-1	
B60A	FF	18	00		STX	TEMP1	
B60D	CE	18	86		LDX	\$\$VV	
B610	FF	18	02		STX	TEMP3	
B613	BD	B8	49		JSR	SUMPAR	

ALIGN

B616	CE 1B 3B	LDX	$\$S1U1M-1$	
B619	FF 18 00	STX	TEMP1	
B61C	CE 18 8A	LDX	$\$UU$	
B61F	FF 18 02	STX	TEMP3	
B622	BD BB 49	JSR	SUMPAR	
B625	CE 18 86	LDX	$\$VV$	
B628	4F	CLRA		
B629	BD BB 8E	JSR	PUSH44	
B62C	CE 18 8A	LDX	$\$UU$	
B62F	4F	CLRA		
B630	BD BB 8E	JSR	PUSH44	
B633	86 0A	LDAA	$\$10D$	
B635	BD 80 00	JSR	MATH	UU*VV
B638	CE 18 82	LDX	$\$UV$	
B63B	4F	CLRA		
B63C	BD BB 8E	JSR	PUSH44	
B63F	CE 18 82	LDX	$\$UV$	
B642	4F	CLRA		
B643	BD BB 8E	JSR	PUSH44	
B646	86 0A	LDAA	$\$10D$	
B648	BD 80 00	JSR	MATH	UV*UV
B64B	86 0C	LDAA	$\$12D$	
B64D	BD 80 00	JSR	MATH	DET=UU*VV-UV*UU
B650	7D 18 5E	TST	KKKK+2	
B653	2F 0B	BLE	LA92	
B655	86 18 5E	LDAA	KKKK+2	
B658	BD BB B4	JSR	PUSH41	
B65B	86 0A	LDAA	$\$10D$	
B65D	BD 80 00	JSR	MATH	DET=DET*(KKKK+2)
B660	CE 18 00	LA92	LDX	$\$TEMP1$
B663	4F	CLRA		
B664	BD BB 68	JSR	PULL4	
B667	CE 18 7A	LDX	$\$VP$	
B66A	4F	CLRA		
B66B	BD BB 8E	JSR	PUSH44	
B66E	CE 18 82	LDX	$\$UV$	
B671	4F	CLRA		
B672	BD BB 8E	JSR	PUSH44	
B675	86 0A	LDAA	$\$10D$	
B677	BD 80 00	JSR	MATH	UV*VP
B67A	CE 18 86	LDX	$\$VV$	
B67D	4F	CLRA		
B67E	BD BB 8E	JSR	PUSH44	
B681	CE 18 7E	LDX	$\$UF$	
B684	4F	CLRA		
B685	BD BB 8E	JSR	PUSH44	
B688	86 0A	LDAA	$\$10D$	
B68A	BD 80 00	JSR	MATH	UV*UP
B68D	86 0C	LDAA	$\$12D$	
B68F	BD 80 00	JSR	MATH	UV*VP-UV*UP
B692	CE 18 00	LDX	$\$TEMP1$	
B695	4F	CLRA		
B696	BD BB 8E	JSR	PUSH44	
B699	86 09	LDAA	$\$9$	
B69B	BD 80 00	JSR	MATH	DELX=(UV*VP-UV*UP)/DET
B69E	24 03	BCC	LA91	
B6A0	7E B7 9E	JMP	LA12	

ALIGN

B6A3	CE 18 72	LA91	LDX	\$DELX	
B6A6	4F		CLRA		
B6A7	BD BB 68		JSR	PULL4	
B6AA	CE 18 8A		LDX	\$UU	
B6AD	4F		CLRA		
B6AE	BD BB 8E		JSR	PUSH44	
B6B1	CE 18 7A		LDX	\$VP	
B6B4	4F		CLRA		
B6B5	BD BB 8E		JSR	PUSH44	
B6B8	86 0A		LDAA	\$10D	
B6BA	BD 80 00		JSR	MATH	UU*VP
B6BD	CE 18 82		LDX	\$UV	
B6C0	4F		CLRA		
B6C1	BD BB 8E		JSR	PUSH44	
B6C4	CE 18 7E		LDX	\$UP	
B6C7	4F		CLRA		
B6C8	BD BB 8E		JSR	PUSH44	
B6CB	86 0A		LDAA	\$10D	
B6CD	BD 80 00		JSR	MATH	UV*UP
B6D0	86 0C		LDAA	\$12D	
B6D2	BD 80 00		JSR	MATH	UU*VP-UV*UP
B6D5	CE 18 00		LDX	\$TEMP1	
B6D8	4F		CLRA		
B6D9	BD BB 8E		JSR	PUSH44	
B6DC	86 09		LDAA	\$9	
B6DE	BD 80 00		JSR	MATH	DELY=(UU*VP-UV*UP)/DET
B6E1	CE 18 76		LDX	\$DELY	
B6E4	4F		CLRA		
B6E5	BD BB 68		JSR	PULL4	
B6E8	FE 18 74		LDX	DELX+2	
B6EB	FF 18 6E		STX	XTRAN	
B6EE	FE 18 78		LDX	DELY+2	
B6F1	FF 18 70		STX	YTRAN	
B6F4	7F 18 90		CLR	ANGLE	
B6F7	7F 18 91		CLR	ANGLE+1	
B6FA	7D 18 72		TST	DELX	
B6FD	2C 06		BGE	LA10	
B6FF	CE 18 72		LDX	\$DELX	
B702	BD BB 36		JSR	COMP	
B705	7D 18 76	LA10	TST	DELY	
B708	2C 06		BGE	LA11	
B70A	CE 18 76		LDX	\$DELY	
B70D	BD BB 36		JSR	COMP	
B710	CE 18 72	LA11	LDX	\$DELX	
B713	4F		CLRA		
B714	BD BB 8E		JSR	PUSH44	
B717	CE 18 76		LDX	\$DELY	
B71A	4F		CLRA		
B71B	BD BB 8E		JSR	PUSH44	
B71E	86 0B		LDAA	\$11D	
B720	BD 80 00		JSR	MATH	A(DELX)+A(DELY)
B723	CE 18 72		LDX	\$DELX	
B726	4F		CLRA		
B727	BD BB 68		JSR	PULL4	
B72A	CE 18 72		LDX	\$DELX	
B72D	4F		CLRA		
B72E	BD BB 8E		JSR	PUSH44	

ALIGN

B731	CE 18 D6	LDX	\$DELTA-2	
B734	FF 18 00	STX	TEMP1	
B737	B6 18 00	LDAA	TEMP1	
B73A	F6 18 01	LDAB	TEMP2	
B73D	FB 18 41	ADDB	L	
B740	89 00	ADCA	#0	
B742	FB 18 41	ADDB	L	
B745	89 00	ADCA	#0	
B747	B7 18 00	STAA	TEMP1	
B74A	F7 18 01	STAB	TEMP2	
B74D	FE 18 00	LDX	TEMP1	
B750	A6 00	LDAA	0,X	
B752	E6 01	LDAB	1,X	
B754	BD BB D1	JSR	PUSH42	
B757	86 0C	LDAA	#12D	
B759	BD 80 00	JSR	MATH	A(DELX)+A(DELAY)-DELTA
B75C	CE 18 76	LDX	\$DELAY	
B75F	4F	CLRA		
B760	BD BB 68	JSR	PULL4	
B763	7D 18 76	TST	DELAY	
B766	2B 36	BMI	LA12	
B768	CE 18 72	LDX	\$DELX	
B76B	4F	CLRA		
B76C	BD BB 8E	JSR	PUSH44	
B76F	86 1C	LDAA	#\$1C	
B771	C6 10	LDAB	#\$10	
B773	BD BB D1	JSR	PUSH42	
B776	86 0C	LDAA	#12D	
B778	BD 80 00	JSR	MATH	A(DELX)+A(DELAY)-7184D
B77B	CE 18 76	LDX	\$DELAY	
B77E	4F	CLRA		
B77F	BD BB 68	JSR	PULL4	
B782	7D 18 76	TST	DELAY	
B785	2A 17	BPL	LA12	
B787	BD A8 20	JSR	STPMTR	TRANSLATE DELX, DELY
B78A	7D 18 55	TST	HOW	
B78D	27 0C	BEQ	LA14	
B78F	7A 18 34	DEC	MODE	
B792	7F 18 38	CLR	FIRST	
B795	BD B7 B2	JSR	FINSCH	DITHER ANGLE
B798	7C 18 34	INC	MODE	
B79B	7E B5 B7	LA14	JMP	TO MEASRE: SAME REGION
B79E	7C 18 41	LA12	INC	
B7A1	B6 18 39	LDAA	NBDRY	
B7A4	B0 18 41	SUBA	L	
B7A7	2D 03	BLT	LA13	
B7A9	7E B5 B7	JMP	LA9	TO MEASRE: NEW REGION
B7AC	86 01	LA13	LDAA	#1
B7AE	B7 ED 15	STAA	CALIBR	
B7B1	39	RTS		
	*	SUBROUTINE FINSCH		
	*	FINE ANGLE SEARCH		
B7B2	CE FF 01	FINSCH	LDX	#\$FF01
B7B5	FF 18 62	STX	ST	
B7B8	CE 00 00	LDX	#0	
B7BB	FF 18 6E	STX	XTRAN	
B7BE	FF 18 70	STX	YTRAN	

ALIGN

B7C1	7D 18 3B	TST	FIRST	
B7C4	2E 06	BGT	FS1	
B7C6	BD B0 40	JSR	MEASRE	
B7C9	BD B8 1D	JSR	MAXSUM	
B7CC	FE 18 8E	FS1	LDX	STEPS
B7CF	FF 18 90		STX	ANGLE
B7D2	70 18 91		NEG	ANGLE+1
B7D5	73 18 90		COM	ANGLE
B7D8	BD A8 20		JSR	STPMTR
B7D8	BD B0 40	FS2	JSR	MEASRE
B7DE	7D 18 6D		TST	SUM+3
B7E1	2E 09		BGT	FS3
B7E3	BD B8 1D		JSR	MAXSUM
B7E6	B6 18 63		LDAA	ST+1
B7E9	B7 18 62		STAA	ST
B7EC	7A 18 63	FS3	DEC	ST+1
B7EF	2B 0E		BMI	FS4
B7F1	FE 18 8E		LDX	STEPS
B7F4	FF 18 90		STX	ANGLE
B7F7	78 18 91		ASL	ANGLE+1
B7FA	BD A8 20		JSR	STPMTR
B7FD	20 IC		BRA	FS2
B7FF	FE 18 8E	FS4	LDX	STEPS
B802	FF 18 90		STX	ANGLE
B805	70 18 91		NEG	ANGLE+1
B808	73 18 90		COM	ANGLE
B80B	7D 18 62		TST	ST
B80E	2E 03		BGT	FS5
B810	2B 07		BMI	FS6
B812	39		RTS	
B813	78 18 91	FS5	ASL	ANGLE+1
B816	79 18 90		ROL	ANGLE
B819	BD A8 20	FS6	JSR	STPMTR
B81C	39		RTS	
	*		SUBROUTINE MAXSUM	
	*		MAX=SUM	
B81D	FE 18 82	MAXSUM	LDX	UV
B820	FF 18 7A		STX	VP
B823	FE 18 84		LDX	UV+2
B826	FF 18 7C		STX	VP+2
B829	FE 18 86		LDX	UV+4
B82C	FF 18 7E		STX	VP+4
B82F	FE 18 88		LDX	UV+6
B832	FF 18 80		STX	VP+6
B835	39		RTS	
	*		SUBROUTINE COMP	
	*		COMPS 4 BYTE NUMBER	
B836	60 03	COMP	NEG	3,X
B838	86 03		LDAA	#3
B83A	25 04	C1	BCS	C2
B83C	60 02		NEG	2,X
B83E	20 02		BRA	C3
B840	63 02	C2	COM	2,X
B842	4A	C3	DECA	
B843	27 03		REQ	C4
B845	09		DEX	
B846	20 F2		BRA	C1

ALIGN

B848	39	C4	RTS	
		*	SUBROUTINE	SUMPAR
		*	SUMS PARTIAL PRODUCTS FOR UP, VP, UV, U, V	
		*	ENTER WITH L, TEMP1-TEMP4	
B849	86 01	SUMPAR	LDAA	#1
B84B	B7 18 37		STAA	BDRY
B84E	B7 18 3A		STAA	BDRYPT
B851	4F		CLRA	
B852	BD BB B4		JSR	PUSH41
B855	FE 18 00	LSM1	LDX	TEMP1
B858	B6 18 3A		LDAA	BDRYPT
B85B	BD BB 8E		JSR	PUSH44
B85E	86 0B		LDAA	#11D
B860	BD 80 00		JSR	MATH
B863	7C 18 37		INC	BDRY
B866	B6 18 41		LDAA	L
B869	B1 18 37		CMFA	BDRY
B86C	2D 0B		BLT	LSM2
B86E	B6 18 3A		LDAA	BDRYPT
B871	8B 04		ADDA	#4
B873	B7 18 3A		STAA	BDRYPT
B876	7E B8 55		JMP	LSM1
B879	B6 18 5C	LSM2	LDAA	KKKK
B87C	BD BB B4		JSR	PUSH41
B87F	86 09		LDAA	#9
B881	BD 80 00		JSR	MATH
B884	FE 18 02		LDX	TEMP3
B887	4F		CLRA	
B888	BD BB 68		JSR	PULL4
B88B	39		RTS	
		*	SUBROUTINE	INIZE
		*	INITIALIZES RAM VARIABLES	
B88C	BF 18 01	INIZE	STS	TEMP2
B88F	CE 18 E2	LIN2	LDX	\$BCDAR
B892	4F		CLRA	
B893	A7 00	LIN3	STAA	0,X
B895	08		INX	
B896	8C 20 00		CPX	#MS1MS+400D
B899	26 F8		BNE	LIN3
B89B	8E 1A E6		LDS	\$BDRYLF+4
B89E	CE B9 34		LDX	\$TABLE+94D
B8A1	A6 00	LIN1	LDAA	0,X
B8A3	36		PSHA	
B8A4	09		DEX	
B8A5	8C B9 2F		CPX	\$TABLE+89D
B8A8	26 F7		BNE	LIN1
B8AA	8E 18 E1	LIN4	LDS	\$DELTA+9
B8AD	CE B9 2F		LDX	\$TABLE+89D
B8B0	A6 00	LINS	LDAA	0,X
B8B2	36		PSHA	
B8B3	09		DEX	
B8B4	8C B9 25		CPX	\$TABLE+79D
B8B7	26 F7		BNE	LINS
B8B9	8E 18 61	LIN6	LDS	\$CORSET1
B8BC	A6 00	LIN7	LDAA	0,X
B8BE	36		PSHA	
B8BF	09		DEX	

ALIGN

B8C0	8C	B8 D5	CPX	\$TABLE-1	
B8C3	26	F7	BNE	LIN7	
B8C5	4F		CLRA		
B8C6	5F		CLRB		
B8C7	CE	18 62	LDX	\$ST	
B8CA	A7	00	LIN9	STAA 0,X	CLEAR UPPER RAM
B8CC	08		INX		
B8CD	5C		INC8		
B8CE	C1	31	CMPB	\$49D	
B8D0	26	F8	BNE	LIN9	
B8D2	BE	18 01	LDS	TEMP2	
B8D5	39		RTS		
B8D6		TABLE	EQU	*	
B8D6	00	00	FCB	0,0	PHASE
B8D8	00	3B E7	FCB	\$0,\$3B,\$E7	X0ZERO=3926
B8DB	F0		FCB	\$F0	
B8DC	00	33 25	FCB	\$0,\$33,\$25	Y0ZERO=3352
B8DF	C0		FCB	\$C0	
B8E0	00	3B C4	FCB	\$0,\$3B,\$C4	X1ZERO=3917
B8E3	C8		FCB	\$C8	
B8E4	00	33 12	FCB	\$0,\$33,\$12	Y1ZERO=3347
B8E7	C8		FCB	\$C8	
B8E8	00	00 00	FCB	0,0,0	COR1 TO BDRYD
B8EB	00	00 00	FCB	0,0,0	
B8EE	00	00 00	FCB	0,0,0	
B8F1	00	00 00	FCB	0,0,0	
B8F4	00	00 00	FCB	0,0,0	
B8F7	00	00 05	FCB	0,0,5	DTIME
B8FA	31	00 00	FCB	49D,0,0	JSTART
B8FD	03	00 00	FCB	3,0,0	NBDRY
B900	00	10 04	FCB	0,\$10,4	NSAMP,LOGS
B903	00	00 00	FCB	0,0,0	
B906	C8	00 00	FCB	\$C8,0,0	LIMIT
B909	00	00	FCB	0,0	
B90B	00	00	FCB	0,0	
B90D	F4	48	FCB	\$F4,\$48	X1INC=-30
B90F	0B	B8	FCB	\$0B,\$B8	Y1INC=30
B911	10	00 00	FCB	\$10,0,0	NPT
B914	00	00 00	FCB	0,0,0	PUSHST,THR11,THR12
B917	00	08 01	FCB	0,8,1	THR21,THR22,HOW
B91A	00	00 00	FCB	0,0,0	IJPTR
B91D	00	10 40	FCB	0,\$10,\$40	FINE,K\$G
B920	00	00 00	FCB	0,0,0	
B923	10		FCB	\$10	KKKK
B924	00	3C	FCB	0,\$3C	CORSE
B926	27	0B 27	FCB	\$27,\$B,\$27	DELTA
B929	0B	00 04	FCB	\$B,0,4	
B92C	00	00 00	FCB	0,0,0	
B92F	00		FCB	0	
B930	31	46 5B	FCB	49D,70D,91D	BDRYLF
B933	5B	5B	FCB	91D,91D	
B935			END		

MEASURE SUBROUTINES

1800	ORG	\$1800		
1800	TEMP1	RMB	1	
1801	TEMP2	RMB	1	
1802	TEMP3	RMB	1	
1803	TEMP4	RMB	1	
1804	TEMP5	RMB	1	
1805	TEMP6	RMB	1	
1806	BCD1	RMB	1	
1807	BCD2	RMB	1	
1808	CTR	RMB	1	
1809	CTR1	RMB	1	
1812	ORG	\$1812		
1812	PHASE	RMB	2	
1814	XOZERO	RMB	4	
1818	YOZERO	RMB	4	
181C	X1ZERO	RMB	4	
1820	Y1ZERO	RMB	4	
1824	COR1	RMB	1	REAL
1825	COR2	RMB	1	
1826	COR3	RMB	1	IMAG
1827	COR4	RMB	3	
182A	COR9	RMB	1	THETA
182B	COR10	RMB	1	
182C	I	RMB	1	
182D	J	RMB	1	
182E	ID	RMB	1	
182F	JD	RMB	1	
1830	UI1	RMB	1	
1831	UI2	RMB	1	
1832	VJ1	RMB	1	
1833	VJ2	RMB	1	
1834	MODE	RMB	1	
1835	DTIME	RMB	1	
1836	JSTART	RMB	1	
1837	BDRY	RMB	1	
1838	BDRYD	RMB	1	
1839	NBDRY	RMB	1	
183A	BDRYPT	RMB	1	
183B	FIRST	RMB	1	
183C	SENSOR	RMB	1	
183D	NSAMP	RMB	1	
183E	LOGNS	RMB	1	
183F	BCDPTR	RMB	2	
1841	L	RMB	1	
1842	LIMIT	RMB	1	
1843	X	RMB	1	
1844	Y	RMB	1	
1845	ZERO	RMB	4	
1849	X1INC	RMB	2	
184B	Y1INC	RMB	2	
184D	NPT	RMB	1	
184E	STACK1	RMB	1	
184F	STACK2	RMB	1	
1850	PUSHST	RMB	1	
1851	THR11	RMB	1	
1852	THR12	RMB	1	
1853	THR21	RMB	1	

MEASURE SUBROUTINES

1854	THR22	RMB	1
1855	HOW	RMB	1
1856	IJPTR	RMB	1
1857	IJPTR1	RMB	1
1858	ADRSW	RMB	1
1859	SETUP	RMB	1
185A	FINE	RMB	1
185B	K\$G	RMB	1
185C	KKKK	RMB	4
1860	CORSE	RMB	2
1862	ST	RMB	2
1864	SAVESP	RMB	2
1866	MAX	RMB	4
186A	SUM	RMB	4
186E	XTRAN	RMB	2
1870	YTRAN	RMB	2
1872	DELX	RMB	4
1876	DELY	RMB	4
187A	VP	RMB	4
187E	UP	RMB	4
1882	UV	RMB	4
1886	UV	RMB	4
188A	UU	RMB	4
188E	STEPS	RMB	2
1890	ANGLE	RMB	2
1892	SMAG	RMB	4
1896	SAVEX	RMB	2
1898	SAVEY	RMB	2
189A	DIRECT	RMB	1
189B	DTHR11	RMB	1
189C	DTHR12	RMB	1
189D	DTHR21	RMB	1
189E	DTHR22	RMB	1
189F	MPLX	RMB	2
18A1	SIGNI	RMB	1
18A2	SIGNQ	RMB	1
18A3	SFLAG	RMB	1
18A4	SNFLAG	RMB	1
18D8	ORG		\$18D8
18D8	DELTA	RMB	10D
18E2	BCDAR	RMB	512D
1AE2	BDRYLF	RMB	5
1AE7	BDRYCT	RMB	5
1AEC	S1VP1M	RMB	20D
1B00	S1UP1M	RMB	20D
1B14	S1UV1M	RMB	20D
1B28	S1V1M	RMB	20D
1B3C	S1U1M	RMB	20D
1B50	IJ	RMB	400D
1CEO	PS1MS	RMB	400D
1E70	MS1MS	RMB	400D
8000	MATH	EQU	\$8000
BDD6	CORDIC	EQU	\$BDD6
DFF0	DAS	EQU	\$DFF0
A5F7	CORD	EQU	\$A5F7
A612	IXST	EQU	\$A612
A619	CALSDI	EQU	\$A619

MEASURE SUBROUTINES

A636		SGNSET	EQU	\$A636
A63D		IDEX	EQU	\$A63D
B9A0			ORG	\$B9A0
	*			
	*			SUBROUTINE DELAY1
	*			GIVES 9.996MS*DTIME DELAY
B9A0	B6 18 35	DELAY1	LDAA	DTIME
B9A3	4D	D1	TSTA	
B9A4	27 06		BEQ	D2
B9A6	BD B9 A0		JSR	DELAY
B9A9	4A		DECA	
B9AA	20 F7		BRA	D1
B9AC	39	D2	RTS	
	*			
	*			
	*			SUBROUTINE DELAY
	*			9.996 MS DELAY
B9AD	CE 02 CA	DELAY	LDX	*2CAH
B9B0	09	LDY1	DEX	
B9B1	27 02		BEQ	LDY2
B9B3	20 FB		BRA	LDY1
B9B5	39	LDY2	RTS	
	*			SUBROUTINE ADDRES
	*			ENTRY USES I,J,SENSOR
	*			REFERENCE SENSOR ORIGIN IN (MS TO LS)
	*			(X0ZERO TO X0ZERO+3),(Y0ZERO TO Y0ZERO+3)
	*			ALIGNING SENSOR ORGIN IN (MS TO LS)
	*			(X1ZERO TO X1ZERO+3),(Y1ZERO TO Y1ZERO+3)
	*			BCD RESULT WILL BE RETURNED IN (MS TO LS)
	*			(UI1,UI2), (VJ1,VJ2)
B9B6	B6 18 3C	ADIRES	LDAA	SENSOR
B9B9	2E 03		BGT	LADD0
B9BB	7E BA 8A		JMP	LADD
B9BE	7D 18 58	LADD0	TST	ADRSW
B9C1	2C 03		BGE	LADD00
B9C3	7E BA 81		JMP	LADD1
B9C6	FE 18 3F	LADD00	LDX	BCDPTR
B9C9	08		INX	
B9CA	BD A6 12		JSR	IXST
B9CD	A6 00		LDAA	0,X
B9CF	B7 18 30		STAA	UI1
B9D2	A6 01		LDAA	1,X
B9D4	B7 18 31		STAA	UI2
B9D7	A6 02		LDAA	2,X
B9D9	B7 18 32		STAA	VJ1
B9DC	A6 03		LDAA	3,X
B9DE	B7 18 33		STAA	VJ2
B9E1	39		RTS	
B9E2	F6 18 2C	LADD3	LDAB	I
B9E5	4F		CLRA	
B9E6	BD BB D1		JSR	PUSH42
B9E9	B6 18 49		LDAA	X1INC
B9EC	F6 18 4A		LDAB	X1INC+1
B9EF	BD BB D1		JSR	PUSH42
B9F2	86 0A		LDAA	*10D
B9F4	BD 80 00		JSR	MATH
B9F7	FE 18 45		LDX	ZERO
				I*X1INC
				TRUE VALUE*.1 (TYPICAL)
				TRUE VALUE*.1

MEASURE SUBROUTINES

B9FA	4F	CLRA		
B9FB	BD BB 8E	JSR	PUSH44	
B9FE	86 0B	LDAA	\$11D	
BA00	BD 80 00	JSR	MATH	ZERO+I*INC
BA03	86 03	LDAA	\$3	
BA05	C6 E8	LDAB	\$\$E8	
BA07	BD BB D1	JSR	PUSH42	
BA0A	86 09	LDAA	\$9	
BA0C	BD 80 00	JSR	MATH	DIVIDE BY 1000
BA0F	CE 18 00	LDX	\$TEMP1	
BA12	4F	CLRA		
BA13	BD BB 68	JSR	PULL4	
BA16	BD BB 17	JSR	BI\$BCD	BINARY TO BCD
BA19	FE 18 06	LDX	BCD1	
BA1C	FF 18 30	STX	UI1	
BA1F	B6 18 2D	LDAA	J	
BA22	BD BB B4	JSR	PUSH41	
BA25	B6 18 4B	LDAA	Y1INC	
BA28	F6 18 4C	LDAB	Y1INC+1	
BA2B	BD BB D1	JSR	PUSH42	
BA2E	86 0A	LDAA	\$10D	
BA30	BD 80 00	JSR	MATH	J*Y1INC
BA33	FE 18 45	LDX	ZERO	
BA36	08	INX		
BA37	08	INX		
BA38	08	INX		
BA39	08	INX		
BA3A	4F	CLRA		
BA3B	BD BB 8E	JSR	PUSH44	
BA3E	86 0B	LDAA	\$11D	
BA40	BD 80 00	JSR	MATH	
BA43	86 03	LDAA	\$3	
BA45	C6 E8	LDAB	\$\$E8	
BA47	BD BB D1	JSR	PUSH42	
BA4A	86 09	LDAA	\$9	
BA4C	BD 80 00	JSR	MATH	
BA4F	CE 18 00	LDX	\$TEMP1	
BA52	4F	CLRA		
BA53	BD BB 68	JSR	PULL4	
BA56	BD BB 17	JSR	BI\$BCD	
BA59	FE 18 06	LDX	BCD1	
BA5C	FF 18 32	STX	UI1	
BA5F	7D 18 58	TST	ADRSW	
BA62	2D 01	BLT	LADD2	
BA64	39	RTS		ADRSW=0
BA65	FE 18 3F	LADD2	BCDPTR	ADRSW=-1 STORE BCD IN ARRAY
BA68	08	INX		
BA69	BD A6 12	JSR	IXST	
BA6C	B6 18 30	LDAA	UI1	
BA6F	A7 00	STAA	0,X	
BA71	B6 18 31	LDAA	UI2	
BA74	A7 01	STAA	1,X	
BA76	B6 18 32	LDAA	UI1	
BA79	A7 02	STAA	2,X	
BA7B	B6 18 33	LDAA	UI2	
BA7E	A7 03	STAA	3,X	
BA80	39	RTS		

MEASURE SUBROUTINES

BA81	CE 18 1C	LADD1	LDX	\$X1ZERO
BA84	FF 18 45		STX	ZERO
BA87	7E B9 E2	LADD4	JMP	LADD3
BA8A	CE 18 14	LADD	LDX	\$X0ZERO
BA8D	FF 18 45		STX	ZERO
BA90	20 F5		BRA	LADD4
	*			
	*			SUBROUTINE READ
	*			READS,STORES REFERENCE DATA
BA92	BD BC 0D	READ	JSR	R1DEFT
BA95	FE 18 24		LDX	COR1
BA98	FF 18 6E		STX	XTRAN
BA9B	FE 18 26		LDX	COR3
BA9E	FF 18 70		STX	YTRAN
BAA1	7F 18 3C		CLR	SENSOR
BAA4	7C 18 58		INC	AIRSW
BAA7	BD B9 B6		JSR	ADRES
BAAA	7A 18 58		DEC	ADRSW
BAAD	BD BB F4		JSR	TUNE
BAB0	BD B9 A0		JSR	DELAY1
BAB3	7A 18 3C		DEC	SENSOR
BAB6	BD BC 0D		JSR	R1DEFT
BAB9	86 01		LDAA	#1
BABB	B7 18 3C		STAA	SENSOR
BABE	7F 18 2A		CLR	COR9
BAC1	7F 18 2B		CLR	COR10
BAC4	BD A5 F7		JSR	CORD
BAC7	CE 1E 6E		LDX	#MS1MS-2
BACA	FF 18 00		STX	TEMP1
BACD	B6 18 00		LDAA	TEMP1
BAD0	F6 18 01		LDAB	TEMP2
BAD3	FB 18 44		ADDB	Y
BAD6	89 00		ADCA	#0
BAD8	FB 18 44		ADDB	Y
BADB	89 00		ADCA	#0
BADD	B7 18 00		STAA	TEMP1
BAE0	F7 18 01		STAB	TEMP2
BAE3	FE 18 00		LDX	TEMP1
BAE6	B6 18 24		LDAA	COR1
BAE9	A7 00		STAA	0,X
BAEB	B6 18 25		LDAA	COR2
BAEE	A7 01		STAA	1,X
BAF0	B6 18		LDAA	TEMP1
BAF3	C0 90		SUBB	#\$90
BAF5	B2 01		SBCA	#1
BAF7	B7 18 00		STAA	TEMP1
BAFA	F7 18 01		STAB	TEMP2
BAFD	FE 18 00		LDX	TEMP1
BB00	B6 18 2A		LDAA	COR9
BB03	A7 00		STAA	0,X
BB05	B6 18 2B		LDAA	COR10
BB08	A7 01		STAA	1,X
BB0A	FE 18 6E		LDX	XTRAN
BB0D	FF 18 24		STX	COR1
BB10	FE 18 70		LDX	YTRAN
BB13	FF 18 26		STX	COR3
BB16	39		RTS	

MEASURE SUBROUTINES

			*	SUBROUTINE	BI\$BCD
			*	BINARY TO BCD	
			*	BINARY IN (MS TO LS) (TEMP3,TEMP4)	
			*	BCD IN BCD1,BCD2	
			*	BINARY NUMBER MUST BE POSITIVE	
BB17	7F 18 06	BB17	CLR	BCD1	
BB1A	7F 18 07		CLR	BCD2	
BB1D	86 10		LDAA	\$16D	
BB1F	B7 18 09		STAA	CTR1	
BB22	BF 18 04		STS	TEMP5	
BB25	8E 18 01		LDS	\$TEMP3-1	
BB28	86 08	LBIB1	LDAA	\$8D	
BB2A	B7 18 08		STAA	CTR	
BB2D	33		PULB		GET ANOTHER BYTE
BB2E	58	LBIB2	ASLB		TEST RIGHT MOST BIT
BB2F	24 12		BCC	LBIB3	
BB31	86 01		LDAA	\$01	ADD 1
BB33	BB 18 07		ADDA	BCD2	
BB36	19		DAA		
BB37	B7 18 07		STAA	BCD2	
BB3A	B6 18 06		LDAA	BCD1	
BB3D	89 00		ADCA	\$0	
BB3F	19		DAA		
BB40	B7 18 06		STAA	BCD1	
BB43	7A 18 09	LBIB3	DEC	CTR1	SKIP DOUBLE IF LAST
BB46	27 1C		BEQ	LBIB4	
BB48	B6 18 07		LDAA	BCD2	DOUBLE
BB4B	BB 18 07		ADDA	BCD2	
BB4E	19		DAA		
BB4F	B7 18 07		STAA	BCD2	
BB52	B6 18 06		LDAA	BCD1	
BB55	B9 18 06		ADCA	BCD1	
BB58	19		DAA		
BB59	B7 18 06		STAA	BCD1	
BB5C	7A 18 08		DEC	CTR	
BB5F	27 C7		BEQ	LBIB1	NEW BYTE
BB61	7E BB 2E		JMP	LBIB2	SAME BYTE
BB64	BE 18 04	LBIB4	LDS	TEMP5	RESTORE STACK
BB67	39		RTS		
		*	SUBROUTINE	PULL4	
		*	PULLS 4 BYTES OFF STACK		
		*	STORES 1ST (MS) BYTE IN ADDRESS X REG+A REG ...		
		*	STORES 4TH (LS) BYTE IN ADDRESS X REG+A REG+3		
BB68	33	PULL4	PULB		
BB69	F7 18 4E		STAB	STACK1	
BB6C	33		PULB		
BB6D	F7 18 4F		STAB	STACK2	
BB70	4D		TSTA		
BB71	27 06		BEQ	LP42	
BB73	08	LP41	INX		
BB74	4A		DECA		
BB75	27 02		BEQ	LP42	
BB77	20 FA		BRA	LP41	
BB79	32	LP42	PULA		
BB7A	A7 00		STAA	0,X	
BB7C	32		PULA		
BB7D	A7 01		STAA	1,X	

MEASURE SUBROUTINES

BB7F	32	PULA		
BB80	A7 02	STAA	2,X	
BB82	32	PULA		
BB83	A7 03	STAA	3,X	
BB85	F6 18 4F	LDAB	STACK2	
BB88	37	PSHB		
BB89	F6 18 4E	LDAB	STACK1	
BB8C	37	PSHB		
BB8D	39	RTS		
*				
*				
*				
* SUBROUTINE PUSH44				
* PUSHES 4 BYTE NUMBER ONTO STACK				
* ADDRESS OF LS BYTE IS X REG + A REG + 3				
BB8E	33	PUSH44	PULB	
BB8F	F7 18 4E	STAB	STACK1	
BB92	33	PULB		
BB93	F7 18 4F	STAB	STACK2	
BB96	4D	TSTA		
BB97	27 06	BEQ	LP442	
BB99	08	LP441	INX	
BB9A	4A		DECA	
BB9B	27 02	REQ	LP442	
BB9D	20 FA	BRA	LP441	
BB9F	A6 03	LP442	LDAA 3,X	
BBA1	36		PSHA	
BBA2	A6 02	LDAA	2,X	
BBA4	36		PSHA	
BBA5	A6 01	LDAA	1,X	
BBA7	36		PSHA	
BBA8	A6 00	LDAA	0,X	
BBA9	36		PSHA	
BBAB	F6 18 4F	LDAB	STACK2	
BBAE	37		PSHB	
BBAF	F6 18 4E	LDAB	STACK1	
BBB2	37		PSHB	
BBB3	39		RTS	
*				
*				
*				
* SUBROUTINE PUSH41				
* PUSHES 4 BYTE NUMBER ONTO STACK				
* 3 MS BYTES ARE SIGN BITS				
* ENTER WITH LS BYTE IN A REG				
BBB4	33	PUSH41	PULB	
BBB5	F7 18 4E	STAB	STACK1	
BBB8	33	PULB		
BBB9	F7 18 4F	STAB	STACK2	
BBBC	36	PSHA		
BBBD	48	ASLA		
BBBE	24 04	BCC	LP411	
BBC0	86 FF	LDAA	#\$FF	NEGATIVE
BBC2	20 01	BRA	LP412	
BBC4	4F	LP411	CLRA	POSITIVE
BBC5	36	LP412	PSHA	
BBC6	36		PSHA	

MEASURE SUBROUTINES

BBC7	36		PSHA	
BBC8	F6 18 4F		LDAB	STACK2
BRCB	37		FSHB	
BBCC	F6 18 4E		LDAB	STACK1
BRCF	37		PSHB	
BRD0	39		RTS	
	*		SUBROUTINE	PUSH42
	*		PUSHES 4 BYTE NUMBER ONTO STACK	
	*		2 MS BYTES ARE SIGN BITES	
	*		ENTER WITH LS BYTE IN B REG, NEXT LS BYTE IN A REG	
BBD1	F7 18 50	PUSH42	STAB	PUSHST
BBD4	33		PULB	
BBD5	F7 18 4E		STAB	STACK1
BBD8	33		PULB	
BBD9	F7 18 4F		STAB	STACK2
BBDC	F6 18 50		LDAB	PUSHST
BBDF	37		FSHB	
BRE0	36		PSHA	
BBE1	48		ASLA	
BBE2	24 04		BCC	LP421
BBE4	86 FF		LDAA	#\$FF NEGATIVE
BBE6	20 01		BRA	LP422
BRE8	4F	LP421	CLRA	POSITIVE
BBE9	36	LP422	FSHA	
BBEA	36		FSHA	
BEBB	F6 18 4F		LDAB	STACK2
BBEE	37		PSHB	
BBEF	F6 18 4E		LDAB	STACK1
BBF2	37		PSHB	
BBF3	39		RTS	
	*		SUBROUTINE	TUNE
	*		TUNES SIGNAL GENERATORS	
BBF4	B6 18 30	TUNE	LDAA	UI1
BBF7	B7 EE 11		STAA	OEE11H
BBFA	B6 18 31		LDAA	UI2
BBFD	B7 EE 10		STAA	OEE10H
BC00	B6 18 32		LDAA	VJ1
BC03	B7 EE 21		STAA	OEE21H
BC06	B6 18 33		LDAA	VJ2
BC09	B7 EE 20		STAA	OEE20H
BC0C	39		RTS	
	*		SUBROUTINE RDDEFT	
	*		STORES RESULTS AS	
	*		REAL: COR1,COR2 IMAG: COR3,COR4	
BC0D	B6 18 3D	RDDEFT	LDAA	NSAMP
BC10	B7 18 08		STAA	CTR
BC13	7D 18 3C		TST	SENSOR
BC16	2B 07		BMI	RDD1
BC18	26 10		BNE	RDD2
BC1A	86 01		LDAA	#1 REF, 1 SAMPLE (SENSOR=0)
BC1C	B7 18 08		STAA	CTR
BC1F	CE 02 03	RDD1	LDX	#\$0203 REF, NS SAMPLES (SENSOR=-1)
BC22	FF 18 9F		STX	MPLX
BC25	7D 18 A4		TST	SNFLAG
BC28	27 06		BEQ	RDD00
BC2A	CE 00 01	RDD2	LDX	#\$0001 ALIGN, NS SAMPLES (SENSOR=1)

MEASURE SUBROUTINES

BC2D	FF 18 9F		STX	MPLX
BC30	CE 00 00	RDD00	LDX	\$0
BC33	FF 18 24		STX	COR1
BC36	FF 18 26		STX	COR3
BC39	FF 18 28		STX	COR4+1
BC3C	7D 18 08	RDD3	TST	CTR
BC3F	27 3E		BEQ	RDD4
BC41	BD A6 19		JSR	CALSDI
BC44	B6 18 26		LDAA	COR3
BC47	BB 18 01		ADDA	TEMP1+1
BC4A	B7 18 26		STAA	COR3
BC4D	B6 18 25		LDAA	COR2
BC50	B9 18 00		ADCA	TEMP1
BC53	B7 18 25		STAA	COR2
BC56	B6 18 24		LDAA	COR1
BC59	B9 18 A1		ADCA	SIGN1
BC5C	B7 18 24		STAA	COR1
BC5F	B6 18 29		LDAA	COR4+2
BC62	BB 18 03		ADDA	TEMP3+1
BC65	B7 18 29		STAA	COR4+2
BC68	B6 18 28		LDAA	COR4+1
BC6B	B9 18 02		ADCA	TEMP3
BC6E	B7 18 28		STAA	COR4+1
BC71	B6 18 27		LDAA	COR4
BC74	B9 18 A2		ADCA	SIGNQ
BC77	B7 18 27		STAA	COR4
BC7A	7A 18 08		DEC	CTR
BC7D	20 BD		BRA	RDD3
BC7F	7D 18 3C	RDD4	TST	SENSOR
BC82	27 3A		BEQ	RDD5
BC84	B6 18 3E		LDAA	LOGNS
BC87	B7 18 08		STAA	CTR
BC8A	7D 18 08	RDD6	TST	CTR
BC8D	27 2F		BEQ	RDD5
BC8F	B6 18 24		LDAA	COR1
BC92	47		ASRA	
BC93	B7 18 24		STAA	COR1
BC96	B6 18 25		LDAA	COR2
BC99	46		RORA	
BC9A	B7 18 25		STAA	COR2
BC9D	B6 18 26		LDAA	COR3
BCA0	46		RORA	
BCA1	B7 18 26		STAA	COR3
BCA4	B6 18 27		LDAA	COR4
BCA7	47		ASRA	
BCA8	B7 18 27		STAA	COR4
BCAB	B6 18 28		LDAA	COR4+1
BCAE	46		RORA	
BCAF	B7 18 28		STAA	COR4+1
BCB2	B6 18 29		LDAA	COR4+2
BCB5	46		RORA	
BCB6	B7 18 29		STAA	COR4+2
BCB9	7A 18 08		DEC	CTR
BCBC	20 CC		BRA	RDD6
BCBE	FE 18 25	RDD5	LDX	COR2
BCC1	FF 18 24		STX	COR1
BCC4	FE 18 28		LDX	COR4+1

MEASURE SUBROUTINES

BCC7	FF 18 26	STX	COR3
BCCA	39	RTS	
BCCF		ORG	\$BCCF
	*	SUBROUTINE INDEX	
	*	INCREMENTS NORMALIZED ADDRESS IN FREQUENCY	
	*	PLANE FOR REFERENCE SEARCH	
	*	ADDRESS IS (I,J) (REAL,IMAG)	
	*	RETURNS INCREMENTED I,J	
	*	USES A,B	
BCCF	B6 18 2C	INDEX	LDAA I
BCD2	4D		TSTA
BCD3	27 03		BEQ IN2
BCD5	7E BC E7		JMP IN3
BCD8	F6 18 2D	IN2	LDAB J
BCDB	5D		TSTB
BCDC	2E 09		BGT IN3
BCDE	86 07		LDAA #7
BCE0	B0 18 2D		SUBA J
BCE3	B7 18 2D		STAA J
BCE6	39		RTS
BCE7	F6 18 2D	IN3	LDAB J
			I=0 AND J>0 OR I NOT=0
BCEA	BD A6 36		JSR SGNSET
BCEI	F0 18 2C		SUBB I
BCF0	82 00		SBCA #0
BCF2	2D 05		BLT IN7
BCF4	2E 17		BGT IN8
BCF6	5D		TSTB
BCF7	26 14		BNE IN8
BCF9	F6 18 2D	IN7	LDAB J
			J-I<=0
BCFC	BD A6 36		JSR SGNSET
BCFF	FB 18 2C		ADDB I
BD02	89 00		ADCA #0
BD04	2D 19		BLT IN10
BD06	2E 0E		BGT IN9
BD08	5D		TSTB
BD09	26 0B		BNE IN9
BD0B	20 12		BRA IN10
BD0D	86 07	IN8	LDAA #7
			J-I>0
BD0F	BB 18 2C		ADDA I
BD12	B7 18 2C		STAA I
BD15	39		RTS
BD16	B6 18 2D	IN9	LDAA J
			I+J>0
BD19	80 07		SUBA #7
BD1B	B7 18 2D		STAA J
BD1E	39		RTS
BD1F	B6 18 2C	IN10	LDAA I
			I+J<=0
BD22	80 07		SUBA #7
BD24	B7 18 2C		STAA I
BD27	39		RTS
	*	SUBROUTINE GSRCH	
	*	SEARCHES 180 KHZ SQUARE IN 30 KHZ INCREMENTS	
BD28	B6 18 2E	GSRSH	LDAA ID
BD2B	27 03		BEQ GSO
BD2D	B0 18 62		SUBA ST
BD30	B7 18 8C	GSO	STAA UU+2
BD33	B6 18 2F		LDAA JD
BD36	B0 18 62		SUBA ST

MEASURE SUBROUTINES

BD39	B7 18 88		STAA	UV+2
BD3C	7F 18 82		CLR	UV
BD3F	7F 18 83		CLR	UV+1
BD42	7F 18 84		CLR	UV+2
BD45	7F 18 85		CLR	UV+3
BD48	B6 18 8C	GS1	LDAA	UU+2
BD4B	BB 18 84		ADDA	UV+2
BD4E	B7 18 2C		STAA	I
BD51	B6 18 88		LDAA	UV+2
BD54	BB 18 85		ADDA	UV+3
BD57	B7 18 2D		STAA	J
BD5A	BD B9 B6		JSR	ADDRES
BD5D	BD BB F4		JSR	TUNE
BD60	BD B9 A0		JSR	DELAY1
BD63	BD BC 0D		JSR	RDDEFT
BD66	7F 18 2A		CLR	COR9
BD69	7F 18 2B		CLR	COR10
BD6C	BD BD D6		JSR	CORDIC
BD6F	B6 18 24		LDAA	COR1
BD72	F6 18 25		LDAB	COR2
BD75	F0 18 67		SUBB	MAX+1
BD78	B2 18 66		SBCA	MAX
BD7B	2C 02		BGE	GS2
BD7D	20 12		BRA	GS3
BD7F	FE 18 24	GS2	LDX	COR1
BD82	FF 18 66		STX	MAX
BD85	B6 18 2C		LDAA	I
BD88	B7 18 8A		STAA	UU
BD8B	B6 18 2D		LDAA	J
BD8E	B7 18 86		STAA	UV
BD91	7C 18 82	GS3	INC	UV
BD94	B6 18 82		LDAA	UV
BD97	B1 03		CMPA	\$3
BD99	27 02		BEQ	GS4
BD9B	20 2B		BRA	GS7
BD9D	7F 18 82	GS4	CLR	UV
BDA0	7F 18 84		CLR	UV+2
BDA3	7C 18 83		INC	UV+1
BDA6	B6 18 83		LDAA	UV+1
BDA9	81 03		CMPA	\$3
BDBA	27 02		BEQ	GS5
BDBD	20 0D		BRA	GS6
BDAF	B6 18 8A	GS5	LDAA	UU
BDB2	B7 18 2E		STAA	ID
BDB5	B6 18 86		LDAA	UV
BDB8	B7 18 2F		STAA	JD
BDBB	39		RTS	
BDBC	B6 18 85	GS6	LDAA	UV+3
BDBF	BB 18 62		ADDA	ST
BDC2	B7 18 85		STAA	UV+3
BDC5	7E BD 48		JMP	GS1
BDC8	B6 18 84	GS7	LDAA	UV+2
BDCB	BB 18 62		ADDA	ST
BDCE	B7 18 84		STAA	UV+2
BDD1	7E BD 48		JMP	GS1
BDD4			END	

CORDIC

			*	REAL IN COR1,COR2 (MSH,LSH)	
			*	IMAG IN COR3,COR4 (MSH,LSH)	
			*	PHASE IN COR9,COR10 (MSH,LSH) (RADIAN/PI)	
			*	IF PHASE=0 RETURNS	
			*	.41169MAG IN COR1,COR2	
			*	PHASE IN COR9,COR10 (RADIAN/PI)	
			*	IF PHASE NOT=0 RETURNS	
			*	.41169(REALCOSTHETA-IMAGSINTHETA) IN COR1,COR2	
			*	.41169(REALSINTHETA+IMAGCOSTHETA) IN COR3,COR4	
			*	USES CORDIC ALGORITHM: 16 ITERATIONS	
1824			ORG	\$1824	
1824		COR1	RMB	1	REAL OR MAG
1825		COR2	RMB	1	
1826		COR3	RMB	1	IMAG OR ANGLE
1827		COR4	RMB	1	
1828		DNUP1	RMB	1	DOWN UP
1829		DNUP2	RMB	1	
182A		COR9	RMB	1	THETA
182B		COR10	RMB	1	
1800		ORG	\$1800		
1800		COR5	RMB	1	RESCALE FACTOR
1801		COR6	RMB	1	
1802		COR7	RMB	1	TEMPORARY X
1803		COR8	RMB	1	
1804		COR11	RMB	1	LOOP CTR
1805		COR12	RMB	1	
1806		COR13	RMB	1	ALPHA ADDRESS
1807		COR14	RMB	1	
1808		COR15	RMB	1	TEMPORARY Y
1809		COR16	RMB	1	
180A		COR17	RMB	1	TEMPORARY THETA MSH
180B		COR18	RMB	1	ZERO
180C		CFLAG	RMB	1	VECTORIZATION OR ROTATION MODE
BDD6		ORG	\$BDD6		
BDD6	86 01	CORDIC	LDAA	#01	
BDD8	FE 18 2A		LDX	COR9	
BDBB	8C 00 00		CPX	#0	
BDEE	27 02		BEG	LCOR0	
BDE0	86 80		LDAA	#80H	ROTATION MODE
BDE2	B7 18 0C	LCOR0	STAA	CFLAG	
BDE5	B6 18 2A		LDAA	COR9	
BDE8	B7 18 0A		STAA	COR17	
BDEB	7F 18 0B		CLR	COR18	NEED A ZERO
BDEE	B6 18 24		LDAA	COR1	
BDF1	F6 18 25		LDAB	COR2	
BDF4	BD BF 50		JSR	SCALE	REAL SCALE FACTOR
BDF7	FF 18 00		STX	COR5	
BDFA	B6 18 26		LDAA	COR3	
BDFD	F6 18 27		LDAB	COR4	
BE00	BD BF 50		JSR	SCALE	IMAG SCALE FACTOR
BE03	FF 18 02		STX	COR7	
BE06	7D 18 01		TST	COR6	
BE09	26 06		BNE	LCOR1	
BE0B	7D 18 03		TST	COR8	
BE0E	26 01		BNE	LCOR1	
BE10	39		RTS		REAL=IMAG=0
BE11	B6 18 01	LCOR1	LDAA	COR6	

BE14	B0 18 03	SUBA	COR8	
BE17	2F 03	BLE	LCOR2	
BE19	7E BE 22	JMP	LCOR3	
BE1C	FE 18 02	LDX	COR7	
BE1F	FF 18 00	STX	COR5	
BE22	B6 18 01	LDAA	COR6	
BE25	B7 18 01	STAA	COR6	RESCALE FACTOR
BE28	CE 00 00	LDX	\$0	
BE2B	FF 18 02	STX	COR7	
BE2E	FF 18 08	STX	COR15	
BE31	FF 18 28	STX	DNUP1	
BE34	B6 18 01	LDAA	COR6	
BE37	4C	INCA		PREVENTS OVERFLOW
BE38	4C	INCA		
BE39	4C	INCA		FIXES 1ST ITERATION
BE3A	B7 18 29	STAA	DNUP2	
BE3D	B6 18 24	LDAA	COR1	
BE40	F6 18 25	LDAB	COR2	
BE43	BD BF 8A	JSR	UP	SCALE REAL UP
BE46	B7 18 24	STAA	COR1	
BE49	F7 18 25	STAB	COR2	
BE4C	B6 18 01	LDAA	COR6	GET RESCALE FACTOR
BE4F	4C	INCA		PREVENTS OVERFLOW
BE50	4C	INCA		
BE51	4C	INCA		FIXES 1ST ITERATION
BE52	B7 18 29	STAA	DNUP2	
BE55	B6 18 26	LDAA	COR3	
BE58	F6 18 27	LDAB	COR4	
BE5B	BD BF 8A	JSR	UP	SCALE IMAG UP
BE5E	B7 18 26	STAA	COR3	
BE61	F7 18 27	STAB	COR4	
BE64	7F 18 04	CLR	COR11	
BE67	86 10	LDAA	\$16D	
BE69	B7 18 05	STAA	COR12	INITIALIZE
BE6C	CE BF 3F	LDX	\$LAGMSH+15	
BE6F	FF 18 06	STX	COR13	ALPHA ADDRESS
BE72	FE 18 06	LCOR4	LDX	MAIN LOOP
BE75	A6 00	LDAA	0,X	LOOK UP ALPHA
BE77	E6 10	LDAB	16,X	
BE79	BD BF EA	JSR	TSTCFL	SIGN OF Y(I-1) OR THETA(I-1)
BE7C	2C 03	BGE	LCOR5	
BE7E	BD BF 6B	JSR	COMP	-ALPHA
BE81	FB 18 2B	LCOR5	ADDB	COR10
BE84	B9 18 2A		ADCA	COR9
BE87	BD BF 7B	JSR	OFLCK	
BE8A	B7 18 2A	STAA	COR9	
BE8D	F7 18 2B	STAB	COR10	THETA DONE
BE90	B6 18 05	LDAA	COR12	
BE93	B7 18 29	STAA	DNUP2	
BE96	B6 18 26	LDAA	COR3	
BE99	F6 18 27	LDAB	COR4	
BE9C	BD BF EA	JSR	TSTCFL	
BE9F	2C 03	BGE	LCOR6	
BEA1	BD BF 6B	JSR	COMP	
BEA4	BD BF BA	LCOR6	JSR	DOWN
BEA7	FB 18 03		ADDB	COR8
BEAA	B9 18 02		ADCA	COR7

BEAD	B7 18 02	STAA	COR7	
BEBO	F7 18 03	STAB	COR8	
BEB3	B6 18 05	LDAA	COR12	
BEB6	B7 18 29	STAA	DNUP2	
BEB9	B6 18 24	LDAA	COR1	
BECB	F6 18 25	LDAB	COR2	
BEBF	BD BF EA	JSR	TSTCFL	
BEC2	2D 03	BLT	LCOR7	
BEC4	BD RF 6B	JSR	COMP	SGNY(I-1)X(I-1)
BEC7	BD BF BA	LCOR7	JSR	DOWN
RECA	FB 18 09	ADDB	COR16	
RECD	B9 18 08	ADCA	COR15	
BED0	B7 18 26	STAA	COR3	
BED3	F7 18 27	STAB	COR4	Y DONE
BED6	B6 18 02	LDAA	COR7	
BED9	F6 18 03	LDAB	COR8	
BEDC	B7 18 24	STAA	COR1	
BEDF	F7 18 25	STAB	COR2	
BEE2	B6 18 26	LDAA	COR3	
BEES	F6 18 27	LDAB	COR4	
BEE8	B7 18 08	STAA	COR15	
BEEB	F7 18 09	STAB	COR16	
BEEE	B6 18 2A	LDAA	COR9	
BEF1	B7 18 0A	STAA	COR17	
BEF4	7A 18 05	DEC	COR12	
BEF7	7A 18 07	DEC	COR14	
BEFA	FE 18 04	LDX	COR11	
BEFI	8C 00 00	CPX	\$0	
BF00	27 03	BEQ	LCOR8	
BF02	7E BE 72	JMP	LCOR4	
BF05	B6 18 01	LCOR8	LDAA	COR6
BF08	B7 18 29	STAA	DNUP2	
BF0B	B6 18 24	LDAA	COR1	
BF0E	F6 18 25	LDAB	COR2	
BF11	BD BF BA	JSR	DOWN	
BF14	B7 18 24	STAA	COR1	
BF17	F7 18 25	STAB	COR2	
BF1A	B6 18 01	LDAA	COR6	
BF1D	B7 18 29	STAA	DNUP2	
BF20	B6 18 26	LDAA	COR3	
BF23	F6 18 27	LDAB	COR4	
BF26	BD BF BA	JSR	DOWN	
BF29	B7 18 26	STAA	COR3	
BF2C	F7 18 27	STAB	COR4	
BF2F	39	RTS		
BF30	LAGMSH	EQU	*	
BF30	00	FCB	0	
BF31	00 00 00	FCB	0,0,0	
BF34	00 00 00	FCB	0,0,0	
BF37	00 00 01	FCB	0,0,1	
BF3A	02 05 09	FCB	2,5,9	
BF3D	12 20 40	FCB	I2H,20H,40H	
BF40	LAGLSH	EQU	*	
BF40	01	FCB	01	
BF41	01 03 05	FCB	01,03,05	
BF44	0A 14 29	FCB	0AH,14H,29H	
BF47	51 A3 46	FCB	51H,0A3H,46H	

CORDIC

BF4A	BB 11 FB		FCB	8BH,11H,0FBH
BF4D	E4 00 00	*	FCB	0E4H,0,0
		*	SUBROUTINE SCALE	
		*	A=MSH, B=LSH	
		*	RETURNS RESCALE FACTOR IN X	
BF50	CE 00 00	SCALE	LDX	\$0
BF53	4D		TSTA	
BF54	2C 03		BGE	LSC1
BF56	BD BF 6B		JSR	COMP
BF59	4D	LSC1	TSTA	CHANGE SIGN
BF5A	27 06		BEQ	LSC2
BF5C	47		ASRA	
BF5D	56		RORB	
BF5E	08		INX	
BF5F	7E BF 59		JMP	LSC1
BF62	5D	LSC2	TSTB	
BF63	27 05		BEQ	LSC3
BF65	54		LSRB	
BF66	08		INX	
BF67	7E BF 62		JMP	LSC2
BF6A	39	LSC3	RTS	
		*	SUBROUTINE COMPLEMENT	
		*	A=MSH, B=LSH	
		*	RETURNS 2'S COMPLEMENT IN A,B	
BF6B	5D	COMP	TSTB	
BF6C	26 06		BNE	LCP1
BF6E	81 80		CMPA	\$80H
BF70	26 02		BNE	LCP1
BF72	C6 01		LDAB	\$01
BF74	43	LCP1	COMA	8000H TO 8001H
BF75	53		COMB	
BF76	CB 01		ADDB	\$01
BF78	89 00		ADCA	\$00
BF7A	39		RTS	
		*	SUBROUTINE OVERFLOW CHECK	
		*	OVERFLOW BY LIMITING	
		*	A=MSH, B=LSH	
BF7B	28 0C	OFLCK	BVC	LOF2
BF7D	4D		TSTA	
BF7E	2C 05		BGE	LOF1
BF80	C6 07		LDAB	\$7H
BF82	C6 FF		LDAB	\$OFFH
BF84	39		RTS	
BF85	86 80	LOF1	LDAA	\$80H
BF87	C6 01		LDAB	\$01
BF89	39	LOF2	RTS	SHOULD BE MINUS DON'T LIKE 8000H
		*	SUBROUTINE UP	
		*	A=MSH,B=LSH, DNUP2=RESCALE FACTOR	
		*	RETURNS A,B SCALED UP 15-DNUP2 PLACES	
BF8A	FE 18 28	UP	LDX	DNUP1
BF8D	B7 18 28		STAA	DNUP1
BF90	B6 18 29		LDAA	DNUP2
BF93	80 0F		SUBA	\$0015D
BF95	B7 18 29		STAA	DNUP2

CORDIC

BF98	B6 18 28		LDAA	DNUP1
BF9B	7F 18 28		CLR	DNUP1
BF9E	7D 18 29		TST	DNUP2
BFA1	2E 0B		BGT	LUP2
BFA3	27 14	LUP1	BEQ	LUP3
BFA5	58		ASLB	
BFA6	49		ROLA	
BFA7	08		INX	
BFA8	8C 00 OF		CPX	\$015D
BFAB	7E BF A3		JMP	LUP1
BFAE	27 09	LUP2	BEQ	LUP3
BFB0	47		ASRA	
BFB1	56		RORB	
BFB2	09		DEX	
BFB3	8C 00 OF		CPX	\$015D
BFB6	7E BF AE		JMP	LUP2
BFB9	39	LUP3	RTS	
		*	SUBROUTINE DOWN	
		*	A=MSH, B=LSH, DNUP2=RESCALE FACTOR	
		*	RETURNS A,B SCALED DOWN 15-DNUP2 PLACES	
BFBA	FE 18 28	DOWN	LDX	DNUP1
BFB0	B7 18 28		STAA	DNUP1
BFC0	B6 18 29		LDAA	DNUP2
BFC3	80 0F		SUBA	\$0015D
BFC5	B7 18 29		STAA	DNUP2
BFC8	B6 18 28		LDAA	DNUP1
BFCB	7F 18 28		CLR	DNUP1
BFCF	7D 18 29		TST	DNUP2
BFD1	2E 0B		BGT	LDN2
BFD3	27 14	LDN1	BEQ	LDN3
BFD5	47		ASRA	
BFD6	56		RORB	
BFD7	08		INX	
BFD8	8C 00 OF		CPX	\$015D
BFD9	7E BF D3		JMP	LDN1
BFDE	27 09	LDN2	BEQ	LDN3
BFE0	58		ASLB	
BFE1	49		ROLA	
BFE2	09		DEX	
BFE3	8C 00 OF		CPX	\$015D
BFE6	7E BF DE		JMP	LDN2
BFE9	39	LDN3	RTS	

* SUBROUTINE TSTCFL
 * IF CFLAG=1 TESTS SIGN OF Y(I-1) (VECTORING MODE)
 * IF CFLAG=80 TESTS SIGN OF THETA(I-1) (ROTATION MODE)
 * RETURNS CONDITION CODE REGISTER STATE

BFEA	7D 18 0C	TSTCFL	TST	CFLAG
BFED	2E 0D		BGT	LTSTC2
BFEF	7D 18 0A		TST	COR17
BFF2	2C 04		BGE	LTSTC1
BFF4	7D 18 0B		TST	COR18
BFF7	39		RTS	
BFF8	7D 18 0C	LTSTC1	TST	CFLAG
BFFB	39		RTS	

CORDIC

BFFC 7D 18 26 LTSTC2 TST COR3
BFFF 39 RTS
C000 END

STATEMENTS =285

FREE BYTES =4864

NO ERRORS DETECTED

GLOSSARY OF ACRONYMS

ACIA	- Asynchronous Communications Interface Adapter
ADC	- Analog-to-digital converter
A/D	- Analog-to-digital
BCD	- Binary-coded decimal
CCTV	- Closed-circuit television
CRT	- Cathode-ray tube
DAS	- Data Acquisition Subsystem
DEFT	- Direct Electronic Fourier Transform
EPROM	- Erasable Programmable Read-Only Memory
LED	- Light-Emitting Diode
LSB	- Least Significant Bit
MSB	- Most Significant Bit
NMI	- Non-Maskable Interrupt
PIA	- Peripheral Interface Adapter
RAM	- Random-Access Memory (read/write)
ROM	- Read-Only Memory
SAW	- Surface Acoustic Wave
UV	- Ultraviolet

AD-A091 840 DEFT LABS INC EAST SYRACUSE NY
IMAGE ALIGNMENT AND CORRELATION SYSTEM. (U)
JUL 80 S E CRAIG, A L MOYER
UNCLASSIFIED 0101-A002

F/6 9/2

DAAK70-78-C-0217

NL

ETL-0237

3 x 3

1000

1

END

DATE

FILED

1-3

DTIC

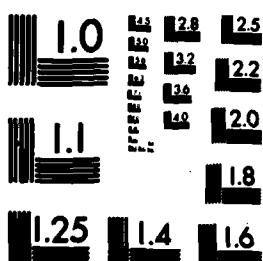
5 OF 3

Distribu

W
U
R
F
A

W
U
S
F
A

A
91840



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Distribution:

W26HAJ **2 cys**

U.S. Army Engineer Topographic Laboratories
Research Institute
Fort Belvoir, VA 22060
ATTN: Joseph F. Hannigan

W26HAJ **14 cys**

U.S. Army Engineer Topographic Laboratories
Survey and Engineering Division
Fort Belvoir, VA 22060
ATTN: STINFO